An Argus Specialist Publication

Computing today

JULY 1984
90p

INCORPORATING MICRO COMPUTER PRINTOUT incorporating BUSINESS MICRO

CONJURING UP A DATA GENIE ON YOUR SPECTRUM

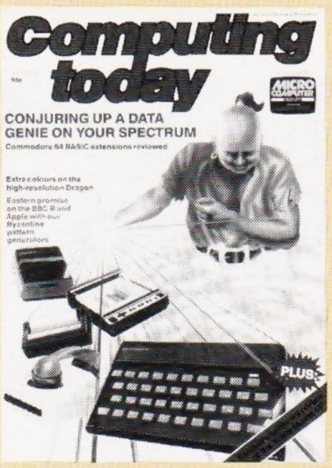
Commodore 64 BASIC extensions reviewed

Extra colours on the high-resolution Dragon

Eastern promise on the BBC B and Apple with our Byzantine pattern generators

PLUS:

Business hardware review of the Tandy Model 4P



CONTENTS

VOL 6 NO 5 JULY 1984

EDITORIAL & ADVERTISEMENT OFFICE
No. 1, Golden Square, London W1R 3AB.
Telephone 01-437 0626. Telex 8811896.

ABC Member of the Audit Bureau of Circulation
ISSN 0142-7210

Computing Today is normally published on the second Friday in the month preceding cover date. Distributed by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS. 01-247 8233. Printed by: Alabaster Passmore & Sons Ltd, Maidstone, Kent.

Subscription rates: UK £13.30 including postage. Airmail and other rates upon application to Computing Today Subscriptions Department, Infonet Ltd, Times House, 179 The Marlowes, Hemel Hempstead, Herts HP1 1BB, England.

Computing Today is constantly on the look-out for well written articles and programs. If you think that your efforts meet our standards, please feel free to submit your work to us for consideration.

All material should be typed. Any programs submitted must be listed (cassette tapes and discs will not be accepted) and should be accompanied by sufficient documentation to enable their implementation. Please enclose an SAE if you want your manuscript returned, all submissions will be acknowledged. Any published work will be paid for.

All work for consideration should be sent to the Editor at our Golden Square address.

NEWS

ACORN MEAN BUSINESS WITH Z80 PROCESSOR

An £299 expansion unit that upgrades the Acorn BBC Microcomputer to a full CP/M-based business computer complete with free business software, was launched by Acorn Computers Ltd last month. Called the Z80 Second Processor, the new upgrade marks Acorn's first step into the serious business market. Tom Hohenburg, Acorn's Marketing Manager, explained: "The Z80-expanded BBC Micro is a low cost yet powerful system. Its full CP/M compatibility opens up a huge range of some 6,000 ready-made applications for the business user, and through our unique 'dual processor' design, it offers a speed and performance better than either competitive expansions or most conventional business systems."

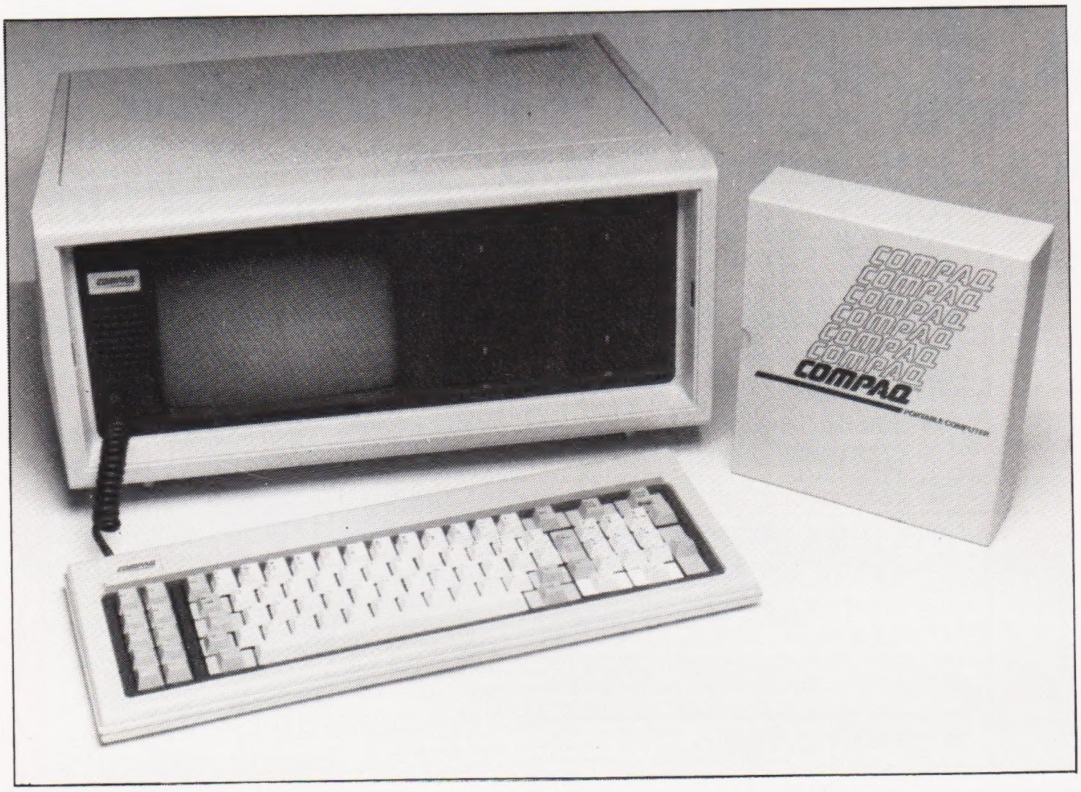
"To support the user of the Z80 Second Processor, we are running special training courses for our dealers, covering all aspects of hardware and software. We consider this vital to ensure that customers gain maximum benefit from this powerful and sophisticated system."

In addition to the Z80 unit and the CP/M2.2 operating system, there is a comprehensive suite of applications programs and development software. The programs have been carefully selected by Acorn from the best available products to provide a complete 'start-up' system for the business and professional home user, meeting the needs of non-programmers and experts alike.

The free software includes three office productivity aids — word processing, database and spreadsheet (with integrated graphics), an integrated accounting system, the award-winning 'Nucleus' applications generator, and the languages CIS COBOL, Professional BASIC and a Z80 version of BBC BASIC.

All software is fully documented, with manuals giving advice for the first time user as well as working instructions. Advice on computerising a business using the Acorn Z80/BBC Microcomputer, and quick reference cards that clip into the BBC Micro's keyboard, are also provided.

The Z80 Second Processor System, complete with unit (in a cream-coloured case matching the BBC Micro), software and manuals, costs £299 including VAT. It is available now from Vector Marketing Ltd, Denington Industrial Estate, Wellingborough, Northants NN8 2RL, and soon from Acorn dealers. Acorn Computers are at Fulbourn Road, Cherry Hinton, Cambridge CB1 4JN.



A COMPACT COMPAQ

The latest addition to the range of personal computers available from MBS is the Compaq Portable. It offers all the advanced features associated with quality personal computers and moreover has the advantage of total transportability. The Compaq is compatible with all the application software, peripherals and expansion devices developed for the IBM PC.

The computer is totally self-contained, the keyboard clips neatly on to the top of the unit for transportation and the complete computer measures only 20" by 8" by 16" when folded. It incorporates a 9" monochrome screen and has a detached typewriter style keyboard.

Memory of 128K RAM is standard, expandable to 640K. 320K of storage is provided by a single diskette drive, with an optional second drive. A 10M fixed disc drive version, the Compaq Plus, is also available having a single 360K diskette drive. A conversion kit can be supplied to upgrade a standard Compaq Plus, complete in every operating function detail.

The 9" screen displays 25 lines by 80 characters with upper and lower case text. The high resolution graphics capabilities enable the creation of clear graphs and charts. The full sized keyboard has the same layout as the IBM and incorporates a numeric keypad and 10 programmable function keys.

There are three slots enabling the computer's performance to be increased by the addition of expansion boards designed for the IBM PC. These cover a wide range and include additional memory, mainframe communications and networking facilities.

Interfaces are provided for a parallel printer, RGB colour monitor, composite video monitor and RF modular interface. All peripheral devices used on the IBM PC can be interfaced to the Compaq.

A rugged construction with a tough outer case ensures that the computer can be transported with total safety. It features a cross-braced aluminium frame that surrounds all the inner components. Specially designed shock mounts suspend the diskette drive to protect it from jolts.

Further details from MBS Personal Computes, 119-120 High Street, Eton, Windsor, Berkshire SL4 6AN (phone 95 68171).

FACTS ON EDFAX

Tecmedia, the Loughborough-based firm known to thousands of teachers as producers of the Input and Micro Primer training packs, has launched its first software product, Edfax, a teletext emulator.

Edfax is a complete simulation of teletext. You can create your own Ceefax lookalike screens or 'pages' of text and graphics, in full colour, store them on disc and recall them (individually, or as a sequence) simply by entering a three digit page number. All you need is a BBC Model B microcomputer with a disc drive, and the Edfax system disc. At present, Edfax is available only on the BBC

RALLYING ROUND WITH KAYPRO

When we first saw this photograph, we thought that some fool was using his computer as a car jack. Not so. Because it's so tough (steel housing and so on), the Kaypro portable computer was chosen to hold all the medical data for the competitors in the Paris-Alger-Dakar off-road rally. So all is now clear: the photo actually shows a doctor providing instant relief to a driver who has inadvertently run himself over.

BASIC TOOLKIT FOR APRICOT AND SIRIUS

Kuma Computers have announced the launch of the Kuma BASIC Toolkit for the ACT Sirius and Apricot micros. The toolkit is a suite of 16 programs designed to aid the development of all types of BASIC programs.

The programs contained in the toolkit can be divided into four categories: general utilities and debugging tools, screen presentation tools, tools for the MSBASIC compiler and demonstration programs. Included with the toolkit is a comprehensive manual and the full source of the demonstration programs to enable easy use of the package.

The Kuma BASIC Toolkit costs £85.00 plus VAT and is available in both Apricot and Sirius formats from Kuma Computers and good micro stores. For further details on this and other Kuma products for the Apricot and Sirius contact Kuma Computers Ltd, 12 Horseshoe Park, Pangbourne, RG8 7JW (phone 07357 4335).

Micro as the program makes full use of the Beeb's graphics capability.

The Edfax master disc includes both display and editing software, together with a demonstration database which provides 80 sample 'pages' full of ideas for using Edfax. These examples are fully discussed in the 88-page users' guide.

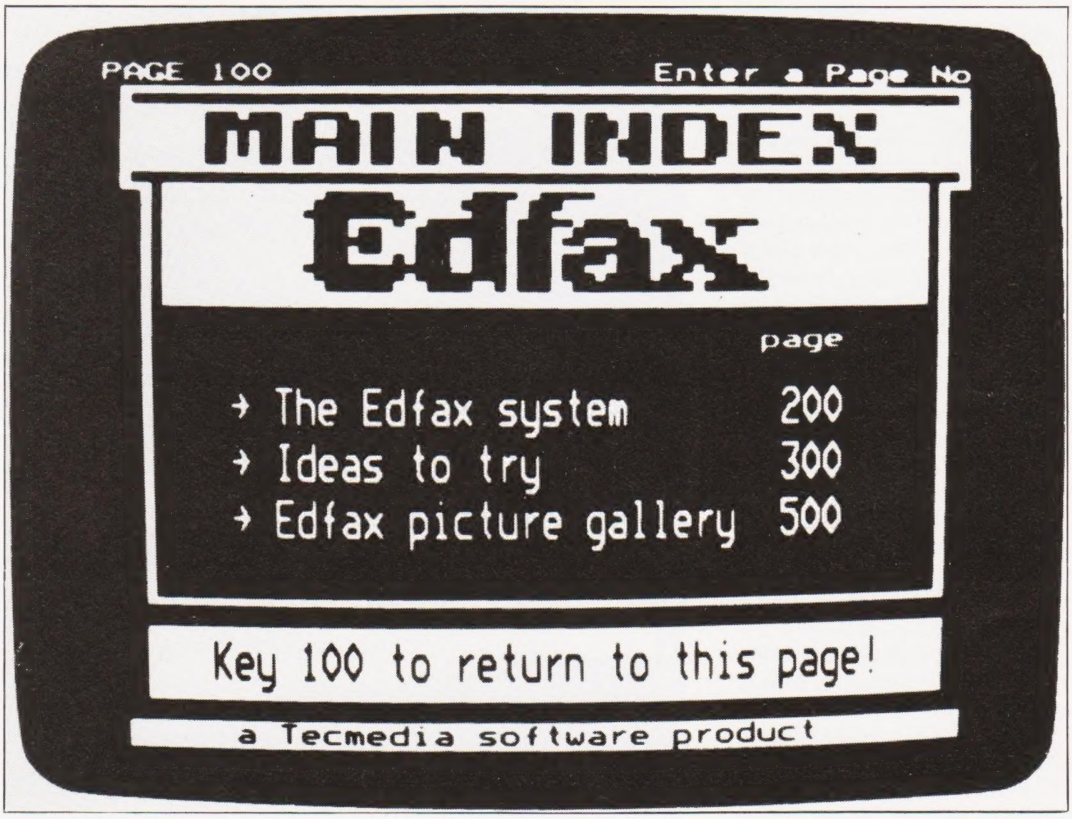
Mike Allen of Tecmedia, who developed the product together with Mike Astill of the Microelectronics Education Programme, says that three things set Edfax apart from its competitors.

"The first is that it is disc based and allows random access to 80 pages of text and graphics on a single 40-track disc drive. Edfax will pack a massive 182 pages on one side of an 80-track disc. This means that users can create really substantial databases.

"The second difference is the uncomplicated nature of the screen editor. Edfax pages can be easily created, even by people who have never touched a computer before.

"Finally, there is substantial documentation, written and edited by educationalists and aimed at the novice user, with the needs of both teachers and students in mind. The language is very straightforward and a number of worked examples show how to create the various effects."

Edfax has been developed over many months, with extensive trials in East Midlands schools. As a result of these trials, 500 copies of the program have been ordered in the East Midlands region alone. Although designed originally for schools use, Edfax is ideal for



use at exhibitions, for retail display and wherever the presentation of information is important.

The Edfax package costs £30 plus £2 post and packing and £2.68 VAT (total £34.68). Schools in the UK get a substantial discount on the basic price and pay only £20 plus £2 post and packing and £1.88 VAT (total £23.88). Technical Data: Edfax requires BBC Model B microcomputer with 40 or 80 track disc drive(s), OS 1.0 (or later) and any Acorn-compatible DFS. Edfax data can be accessed through an Econet (Level II).

Edfax is available from Tecmedia Ltd, 5 Granby Street, Loughborough LE11 3DU (phone 0509 230248).

CRACKING "THE CODE" FOR CASH

A new era of computer games is about to begin with The Code from Soft Concern Ltd of Warwick. The Code is an adventure game with a difference: challenging and unique, with the added incentive of a cash or computer equipment prize to the value of £2,500 to the first correct entry. The price of The Code is £9.95, it runs on the Sinclair Spectrum (48K) and is only available by mail order from The Code, 49 Albert Street, Warwick. No copies will be released until July 1st to give everyone an equal chance to win.

The Code has all the essential ingredients of a spy thriller. You find yourself in a Russian secret military establishment constructed on four levels with limited time to crack the clues necessary to move between levels and get nearer the Code Room where the real puzzle begins. There are enemy agents to avoid or kill, time bombs planted that you must defuse, red herrings and a whole lot more to keep the computer gaming community at work for hours. Plus, of course, the added incentive of £2,500 for the first person to crack The Code and 10 runner-up prizes of £25.

For security all copies of the cassette will be coded along with the buyer's name and address. All cheques sent must have a banker's card number on the reverse or it will have to be returned. Postal orders are welcome.

NEW MANUAL

If you recall the review of the Spectravideo SV-318 in the April issue of *Computing Today,* you know that the manual came in for fairly heavy criticism. In a word, it was awful. Since then we have received a copy of the new, improved version which is supplied with all new machines and are happy to report that it makes a lot more sense. It's still not ideal — you appear to have to work out how to use the joystick from a single example program which contains two errors, for example — but it's no worse than many other manuals.

RELATIVELY INTERESTING . . .

Tatung (UK) Limited has announced the release of its new low-cost, high performance microcomputer which combines the best attributes of an advanced home computer with many of the sophisticated features of a large office computer. It is hoped to provide a unique capability for home, small business and educational users.

Priced at £499, the Einstein was conceived and developed for a specific sector of the market including the enlightened home computer user wanting to make greater use of his new-found interest, the educationalist, and the small business user.

The Z80A microprocessor, running at its maximum speed of 4 MHz, was chosen as the CPU because it is "well-proven, well-liked and well-understood". Additionally there is a wide range of books about the Z80, and a considerable wealth of machine code programs for users to draw from.

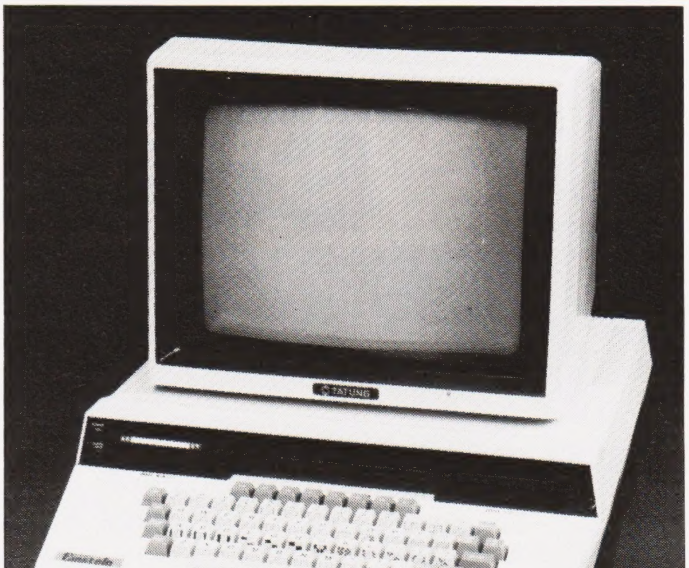
Although there are fast microprocessors on the market, Tatung felt that the additional speed was not warranted, the resulting savings being better invested elsewhere in the design.

There are two common limitations with home computer memory. One is its small size, the other is that the display takes up some of the already small memory. Einstein's advanced display generator has its own 16K of memory, so that full use can be made of the extensive colour and graphics, while allowing full access to the 64K RAM for programs.

56 alphanumeric characters and 160 graphics symbols can be generated on the Einstein, all of which can be reprogrammed by the user. The text format chosen is 40 columns by 24 rows, as it is the standard used by broadcasting and telecommunication authorities for Teletext and Viewdata. This standard was established to get the best text display on a normal domestic colour TV tube. Text can also be displayed in 32 columns and 16 colours.

As graphics are becoming increasingly demanded by the small business user and achieving increasing popularity with the home user, Tatung decided that sophisticated graphics and a colour display was an essential part of the new microcomputer. Einstein, therefore, uses the Texas Instruments display processor chip which currently leads the field in providing 33 different display planes on which to program up to 32 sprites. The graphics display has a resolution of 256 pixels horizontally and 192 pixels vertically, with 16 colours.

The Einstein keyboard has 67 keys, comprising graphics keys, 8 user-programmable special function keys giving 16 functions for the user's own specific applications and 11 control and cursor movement keys.



Built-in features of the Einstein also include a disc drive, thereby avoiding the extra cost of a separate unit with its own power supply.

Einstein uses Teac supersilent 3-inch compact floppy disc drives. They are housed in 'damage proof' hard plastic covers and are small, easy to store and need not be as carefully handled as the conventional floppy disc. The construction of the disc housing makes it virtually 'child proof'. These discs are already widely available from leading High Street software/computer stockists but certified diskettes under the Tatung label will also be made available from all Einstein stockists.

Einstein's disc operating system was specially developed by Crystal Research Ltd and is known as Tatung/Xtal DOS. Unlike many other disc operating systems, Tatung/Xtal DOS has been developed with the first-time user in mind, providing powerful facilities and utilities, but having user-friendly commands and error messages.

For the business user, Tatung/Xtal DOS already has the ability to run CP/M programs, providing for fast sequential and random access file handling and will handle compressed, secret and ASCII files. It also allows full interaction between all files and input/output ports.

The BASIC used is Tatung/Xtal BASIC 4. An enhanced version of the widely accepted Xtal BASIC 3, developed by Crystal Research, it provides over 190 commands and functions, many of which provide for

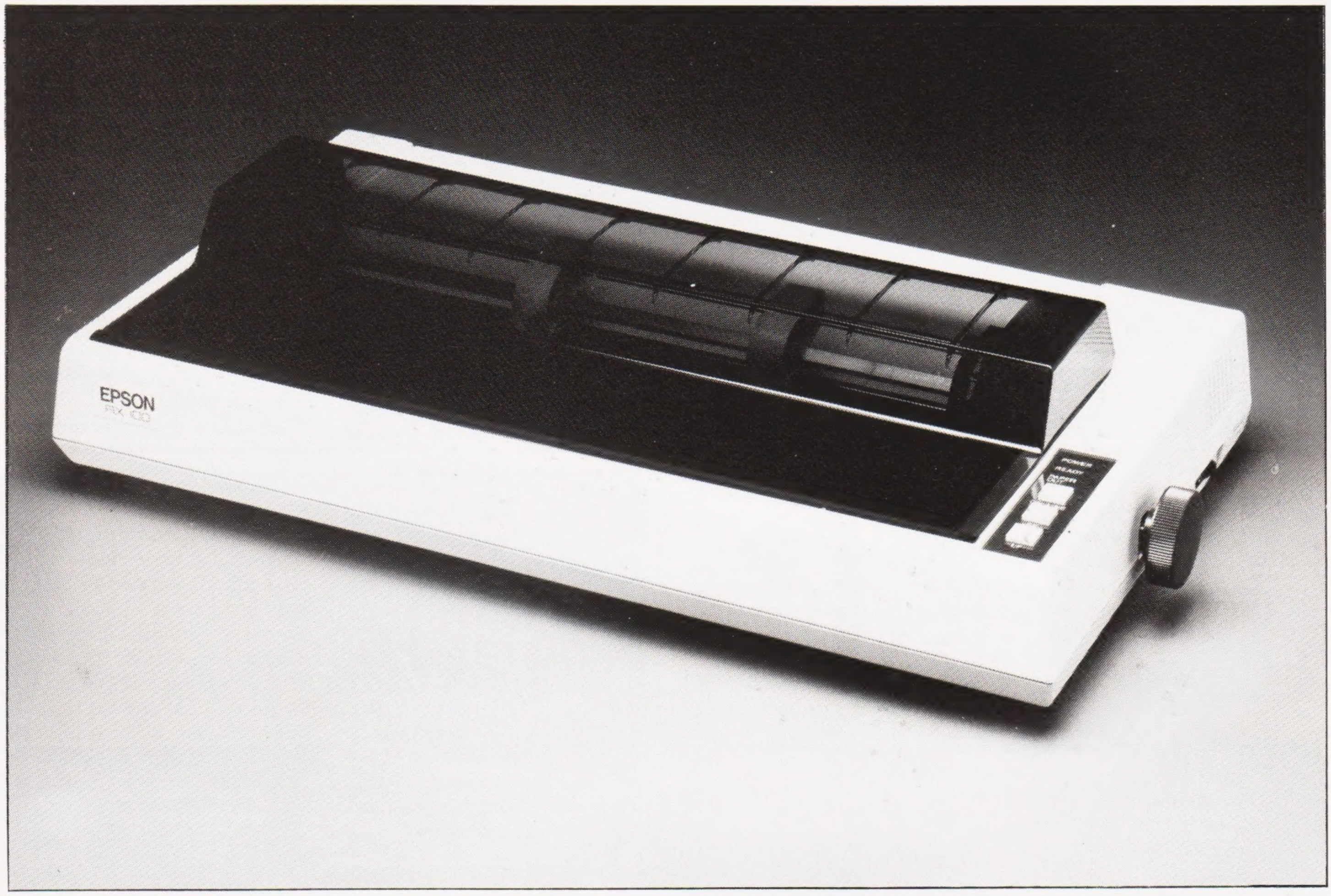
IT'S EPSON AGAIN

Epson (UK) Ltd have announced that a new wide carriage dot matrix printer, model RX-100, is now available in the UK market. The RX-100 is a replacement for the successful MX-100 printer, but with greater versatility, improved styling and a higher print quality. However, new automated production techniques at their Japanese manufacturing plants have enabled Epson to reduce the printer cost by just over 5% to the MX-100.

The RX-100 incorporates all the features of the proven RX-80 F/T with the additional benefit of a wide, 15 inch carriage. Standard print speed is 100 characters per second. A half-speed facilitiy is also incorporated for applications requiring reduced noise. Other features include tractor and friction feed, two full 96 ASCII character sets, 11 international character sets, and 32 graphics characters.

A new 9 by 9 dot matrix head gives improved print definition and graphics capability. Other improvements include proportional spacing and additional Elite and Pica type sizes. The RX-100 will retail at £450 plus VAT.

For more information contact Epson (UK) Limited, Dorland House, 388 High Road, Wembley, Middlesex HA9 6UH (phone 01-902 8892: telex 8814169).



powerful graphics, sprite and sound handling ability.

Another very important feature of Einstein is its wide range of integral input-output facilities. For two-way general data communication, a serial RS232-C port allows connection to other computers or connection via a modem to a telephone link for Viewdata. Einstein also includes a high speed 4-channel, 8-bit resolution analogue to digital converter, so that continuously variable commands, as for example from a pair of joysticks, strain gauges or a pyrometer, can be handled. Bar code wands and graphics tablets can also be connected.

A parallel output to Centronics interface standard permits other ancillaries, such as a printer, to be connected. There is also an 8-bit user port, with 2 control signals. There are two outputs to the display unit. The main output is an analogue video YUV signal chosen in preference to the more conventional analogue RGB signal because it permits adjustment of colour saturation. The YUV signal can, if required, be changed to RGB plus separate sync by means of internal links. A UHF output is also provided, allowing Einstein to be connected to a domestic television receiver. All the Z80A signals are buffered to TTL levels and wired to a 60-way connector.

To provide sound, Einstein uses the GI sound generator which has three channels, plus noise. Each channel can be independently programmed in both pitch and amplitude, providing a wide variety of music and sound effects. Easy-to-use commands in BASIC allow full control of the sound generator. Tatung have used the switch mode technology already incorporated in their range of colour television receivers to "ensure a cool running unit with high reliability". Extensive filtering and care in the design of the power supply has, hopefully, minimised the risk of data corruption due to mains borne interference.

There are several software houses already supporting the Einstein including Crystal Research, Kuma, A & F Software, Leasalink Viewdata, Gemsoftware and Microsimplex. The range covers home games, education, small business, hobby and specialised uses. A wide range of languages can be supported by Einstein such as FORTH, Logo, Pascal, CBASIC, COBOL, FORTRAN and Assembler.

There is a 14" colour display unit available but it is designed to suit a variety of computers in addition to the Einstein. It can accept both YUV and RGB plus sync, switching automatically between the two. These linear systems anticipate the future use of 256 colours. Other peripheral equipment includes printers, joysticks and a twin disc drive unit giving, together with the two internal drives, a total capacity of 2Mb. Tatung hope to add other items such as Prestel operation and 80 column colour display.

Further details from Tatung (UK) Ltd, Pale Meadow Works, Hospital Street, Bridgnorth WV15 6BQ. Tel: Bridgnorth (07462) 5721.

YOUR STARTER FOR TEN

To follow in the footsteps of The French Mistress, The German Master and The Spanish Tutor, Kosmos Software have released the first in a new series of educational programs for the BBC and Electron. The first, titled Answer Back, covers General Knowledge for the over-elevens.

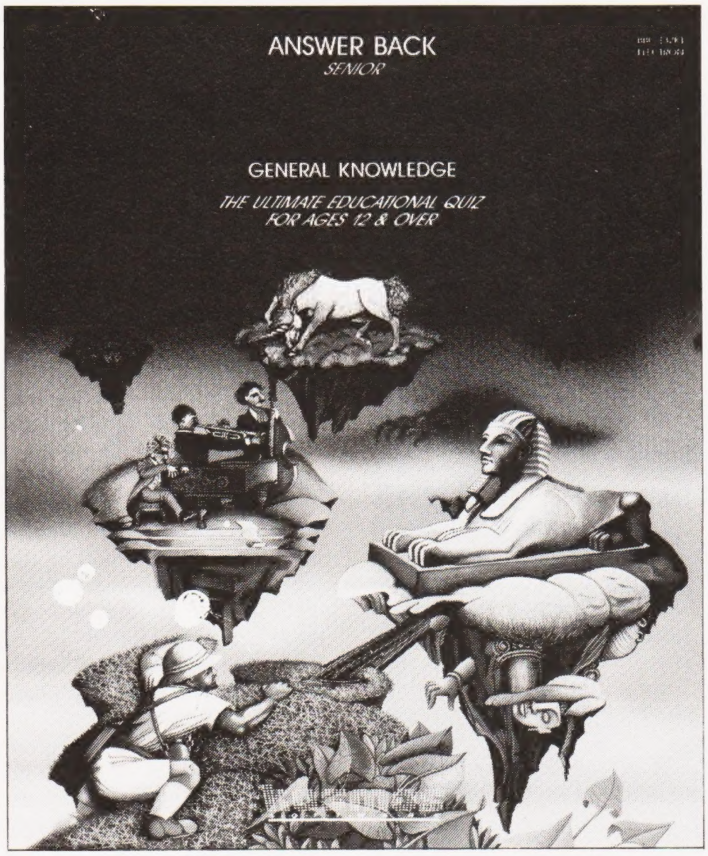
Kosmos state that it has taken six months to develop and it combines "a massive series of well-researched quizzes on General Knowledge topics with a colourful and compelling Space-age game." A sort of Mastermind meets Megamania — the idea being that it will be both instructive and enjoyable to use. Fifteen quizzes are supplied on the cassette covering subjects such as Science, History, Goegraphy, Astronomy, Music, Sport and a host of others.

Once the chosen quiz has been loaded from the master program, simple commands enable it to be used in a variety of different ways including Multiple Choice, True or False and Complete the Answer. The questions are fired onto the screen by a friendly robot armed with a laser gun and correct answers are rewarded with a chance to destroy a fast moving, alien space ship with the robot's gun. Challengers can "pass" questions which they cannot answer and incorrect responses result in the correct answer being displayed immediately below the question.

At the end of the quiz a results summary is displayed. This indicates the time taken, the number of questions correctly answered, incorrectly answered and "passed".

The quizzes themselves are serious and educational containing a balanced mix of questions covering a host of fascinating topics. A total of 750 questions with 3000 answer options are available but challengers tell the computer how many they want and on what topic. The master control program also includes all the necessary facilities to enable an unlimited number of new, multiple-choice quizzes to be created, edited and saved for later use.

The Answer Back quiz, priced at £10.95, is available for both the BBC and Electron, although Kosmos plan to release a version for the Spectrum shortly, along with a General Knowledge for Juniors package for the under-elevens. Further details can be obtained from Kosmos at 1 Pilgrims Close, Harlington, Dunstable, Beds LU5 6LX. Tel: Toddington (05255) 3942.



SHOPPING AT STEIGER

At Stonebridge Park on the North Circular Road, Steiger Computers Ltd has opened the world's largest computer store. 25,000 square feet of showroom houses most of the world's leading computer names and products. Customers, both the home computer user and the business executive have access to a vast range of computers, software, books, magazines and information. The Steiger Home Computer centre offers an enormous range of both educational and game software for youngsters and indeed the whole family. Prices are very competitive and Steiger will be running in-store offers on a regular basis.

The philosophy behind the store is explained by John Patterson, director/general manager of Steiger: "The aim of Steiger is to offer the right product to the right person. Steiger is something very different in the microcomputer market place. For the first time a customer can have everything he needs to set up his computer system, large or small, *to hand,* under one roof: from the smallest computer up to a dedicated multi-user system.

"We support the fact that we have the biggest showroom, and the widest range of products, with skilled people who can give sound, unbiased advice. We offer a programme of after sales training, support and maintenance that is second to none. Here a customer can come and seek advice in comfort, decide in comfort and buy in comfort safe in the knowledge that he can return as often as he wishes.

"In the long term, it is our objective to open a chain of Steiger-owned or franchised shops and to achieve nationwide coverage".

Steiger is the only company

BIG BACKUP

Micro Memory Systems Ltd (MMS), the Newbury-based specialists in data storage systems for micros, has launched a stand-alone cartridge tape unit which provides up to 60 Mbytes of data backup capacity for IBM-XT, Sirius and Apricot systems.

Based on the Wangtek ¼" tape streamer, the unit will back up a 20 Mbyte Winchester in just four minutes, or a 45 Mbyte disc in nine minutes. It uses interchangeable tape cartridges, with a data transfer rate of 90 Kbytes per second at a tape speed of 90 ips (inches per second).

Measuring just 42 by 19 by 14 cm, the tape unit is completely self-contained, needing only a mains power supply and link to the host microcomputer. For more details contact Micro Memory Systems Ltd, Kennet House, Newbury, Berkshire RG13 1JN (phone 0635 40405: telex 848277).

in the UK offering specific and tailor-made disaster recovery services to mainframe users. Their VAX recovery service is offered on two scales; primary, which guarantees access in the event of a disaster, and secondary which offers access on a time basis with day-to-day access. Steiger also intends to become involved in software development, VAX training and systems sales and it is anticipated that this side of the business will keep pace with the growth of the overall Steiger operation.

Towards the end of 1984 Steiger will concentrate on making inroads into the trade marketplace and intends to become established as a major distributor in all areas including printers, monitors, disc drives and all other peripheral products. Steiger is at Steiger House, North Circular Road, Stonebridge Park, London NW10 7QZ (phone 01-961 6000).

THE PRINT PANTHER

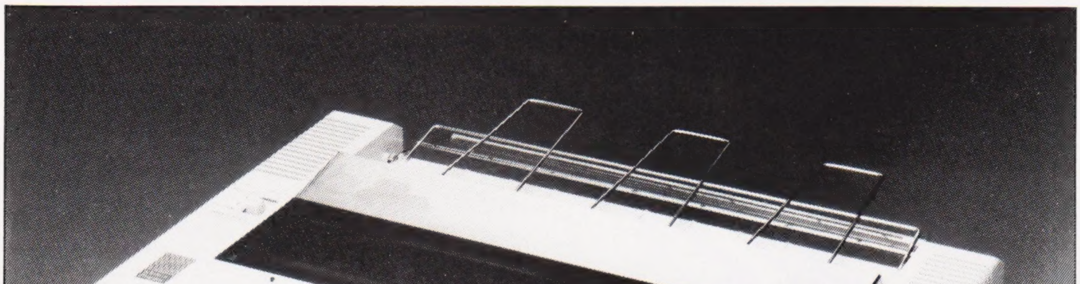
Viewdata character set as standard; double printing facility; 1152 bytes; exceptionally long ribbon life (4 million characters); bi-directional logic seeking head; friction and tractor feed; 1:1 ratio graphics; £278 end-user price. All these are standard on the new Panther printer from Datac.

State-of-the-art engineering and heavyweight design make the Panther ideal for business and home use. It is compatible with the BBC Micro and interfaces with most micro and personal computers. The sophisticated microprocessor-based electronics ensures the versatile and reliable performance that has become a hallmark of all Datac products. Graphic modes include a bit image and high resolution and a 1:1 aspect ratio, plus Viewdata and Teletext block graphics with some customer definable specified characters as standard.

The Panther is bi-directional and logic seeking. It prints pica and elite, and upper and lower case alphanumeric characters. It will also underline, compress and elongate characters. Paper transport is by adjustable friction feed or tractor feed and can accommodate paper widths from 4" to 10". It can print up to three clear copies at a time from an easily changeable cartridge ribbon. The black seamless ribbon has a 4,000,000 character life.

A precision print head utilises special wires for sharp, quality printing and durability and an eight-bit parallel with a true one line 1152 byte input buffer allows for efficient data transfer, making space for a complete line of high resolution pin-addressable graphics.

For further information contact Datac Ltd, Tudor Road, Altrincham, Cheshire WA14 5TN (phone 061-941 2361: telex 667822 CHACOM G).



BOOKING UP ON THE 64

Two more books join the ever-expanding collection of literature dedicated to the Commodore 64, this time hailing from Hayden. The first is **Commodore 64 Programs For the Home** by Charles D. Sternberg, containing listings for home finances, scheduling, schoolwork and so on. The second, **Basic Commodore 64 BASIC,** is another contender for the "we can explain the machine better than the manual" award. (In fact anyone can explain the machine better than the manual!) The book gives a thorough treatment of the subject. There are no prices on the books, but if you harangue your local book merchant the information should be forthcoming.

Small business can now stop going by the book.

For under £1,000 a small business can now equip itself with a BBC Microcomputer, a disc drive, a word processor and printer.

(All tax deductible by the way.)

Once you've parted with that money, you'll find that business has never been brisker.

Because now, there's a new series of floppy disc software specially for the smaller business.

It has been developed by Acornsoft, the software division of Acorn Computers who are the manufacturers of the BBC Micro.

For only £24.95, each disc can store volumes of vital bookwork which can be updated and amended in a fraction of the conventional time.

And there is a disc to cover most aspects of paperwork and book-keeping.

The Invoicing package.

This program stores details of products, VAT numbers and, of course, the names and addresses of your customers. As orders come in, you simply record them. Then, when it's time to invoice, you just press a few keys and each invoice or credit note is printed automatically in seconds.

Allowing for variable terms of trading, the system calculates and prints discounts. And it should help to improve your cash flow dramatically.

The Order Processing package.

With this program, you can confirm your customers' orders, prepare and print despatch notes and make fast analyses of individual orders or of all the orders stored on disc.

The Accounts Receivable package.

Now, it couldn't be easier to keep your customer accounts under control.

In an instant, you can analyse debtors, produce statements, keep a check on any credit limit and calculate VAT output automatically.

Using this package in conjunction with the invoicing package, you can also keep tabs on payments received against payments outstanding.

The Accounts Payable package.

This package will keep you fully up-to-date on how much you owe and who to. In addition, it calculates

Subscriptions

Personally, we think you'll like our approach to microcomputing. Each month, we invite our readers to join us in an abundance of feature articles, projects, general topics, software listings, news and reviews — all to help committed micro users make more of their microcomputers at home or at work.

However, if you've ever missed a copy of Computing Today on the newstands, you'll not need us to tell you how valuable a subscription can be. Subscribe to CT and for a whole year you can sit back, assured that each issue, lovingly wrapped, will find its way through your letter box.

And it's not difficult! All you have to do is fill in the form below, cut it out and send it (or a photocopy) with your cheque or Postal Order (made payable to ASP Ltd) to:

COMPUTING TODAY Subscriptions,

Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.

Alternatively, you can pay by Access or Barclaycard in which case, simply fill in your card number, sign the form and send it off. Please don't send in your card.

Looking for a magazine with a professional approach with material written by micro users for micro users? Why not do yourself a favour and make 1984 the year you subscribe to Computing Today and we'll give you a truly personal approach to microcomputing.

SUBSCRIPTION ORDER FORM

Cut out and SEND TO :
COMPUTING TODAY Subscriptions
INFONET LTD,
TIMES HOUSE,
179 THE MARLOWES,
HEMEL HEMPSTEAD,
HERTS HP1 1BB.

Please commence my subscription to Computing Today with the .... issue.

SUBSCRIPTION RATES (tick ☐ as appropriate)

- £13.30 for 12 issues UK ☐
- £16.95 for 12 issues Overseas Surface ☐
- £36.60 for 12 issues Overseas Air Mail ☐

I am enclosing my (delete as necessary) cheque/ Postal Order/ International Money Order for £..........
(made payable to ASP Ltd)
or
Debit my Access/ Barclaycard*
(*delete as necessary)

BARCLAYCARD VISA

Please use BLOCK CAPITALS and include postcodes.

NAME (Mr/ Mrs/ Miss) ....
delete accordingly

ADDRESS ....

....

.... POSTCODE ....

Signature ....

Date ....

CT July '84

NEXT MONTH

computing today

AUGUST ISSUE ON SALE JULY 13th

We have a host of wonderful features for you in the August issue of *Computing Today*. There's a hardware review of the machine our reviewer has been itching to get his hands on, the Apricot xi with hard disc. BBC owners get two listings, one for a teletext editor, the other for compressed, readable text in Modes 2 and 5. There are two new string commands for the Commodore 64, and a review of the Adventures from Level 9, the wizard of text-compression. Don't miss next month's *Computing Today*.

GIVE THIS TO YOUR NEWSAGENT

Please reserve me a regular copy of Computing Today

NAME ....

ADDRESS ....

POSTCODE ....

Articles described here are in an advanced state of preparation but circumstances may dictate changes to the final contents.

COMMODORE BASIC EXTENSIONS

Jamie Clary

DIY programmers are, no doubt, following CT's series on extending the 64's BASIC. But for those of you who prefer the prepackaged product to the handcrafted variety we have taken an in-depth look at some of the best packages available.

Although most people acknowledge that the Commodore 64 is a popular machine, it is generally recognised that the 64 did not reach its current status by virtue of the quality of its resident BASIC. As all owners would agree, the 64's graphics, sprite and sound generation facilities are impressive but a legacy from the PET era has given it a language which is, by modern standards, very inadequate. Users will not need to be reminded that Commodore BASIC possesses no commands which can directly exploit the Hi-Res graphics and sound facilities and the the only way to manipulate these features from BASIC is by POKEing values directly into screen-memory. Sure enough, Commodore, shortly after launching the 64, had the audacity to release SIMONS' BASIC, a package designed to compensate for these shortcomings by adding 'the 114 missing commands'. However, this did not appease the masses. A price-tag in excess of £50 put SIMONS' beyond the reach of many, and tales that SIMONS' owners were still left wanting, deterred others. This helped fire the imaginations of enthusiastic 64 programmers who promptly discovered ways to endow their micros with a multitude of new, BASIC commands — hence our very own 'Extending The 64's BASIC' series. A few of these 'second-generation' Commodore extension packages surfaced as marketable products and soon, several important rivals to SIMONS' BASIC appeared.

But what advantages, if any, can the new generation offer? To answer this, let us first take a look at the father of these packages, Commodore's very own SIMONS' BASIC.

SIMONS' BASIC

Like all the other language extension packages, SIMONS' BASIC recognises that Commodore's original language is severely deficient in several key areas but particularly in its treatment of sound and graphics. In addition to catering for these very important areas, though, the designer of SIMONS' (one David Simons, I understand) has also added a multitude of supplementary commands which altogether simplify the process of writing programs.

● **Programming Aids** Defining the function keys, something which Commodore completely omitted to mention in the 64 manual, has been catered for quite adequately in SIMONS'. KEY n, "<code>", for example, is all one would need to set up a function key, whilst the DISPLAY command reveals the present key definitions. Other aids, such as automatic line-numbering and renumbering, have also been added but much to my dismay, the automatic line-numbering facility does not have a default setting (most other AUTO commands do!) and the RENUMBERing command cannot resolve GOTO or GOSUB statements. In theory this is fine, because as Commodore like to emphasise in the manual, SIMONS' possesses commands which make GOTOs virtually obsolete. However, this does put constraints upon the use of the RENUMBER facility and such constraints need not exist.

Temporarily suspending the execution of a program can often be of great assistance when attempting to debug a program and David Simons, in recognition of this, has included a PAUSE command to the language. This command follows the syntax PAUSE "<message>", n where <message> can be any string (including a null) and n is the duration of the pause in seconds.

In order that SIMONS' BASIC commands can be recognised in a program, a facility for highlighting all such commands has been included. The OPTION statement displays SIMONS' commands in reverse-video and greatly simplifies all attempts to spot the 'special' commands.

The FIND command causes the 64 to print out all occurrences of a specified string or piece of code and DUMP, an excellent addition, prints the identifiers and associated values of all variables used within a given program, onto the screen.

A very interesting program-security function has also been included. By preceding executable BASIC code by the DISAPA statement, the code will remain visible until the SECURE command is given. This command renders all BASIC statements which follow a DISAPA, invisible when listed. Note: a command for releasing the SECUREd portions of a program is not included!

● **Input Validation** Several advanced string manipulation commands have been added, as have a number of 'input validation' features. INSERT allows a string to be inserted into an existing string, with the existing string 'opening-up' to accept the new characters, whereas INST performs a similar operation but the new characters are superimposed upon the old. CENTRE is another, useful, text-formatting command which, as should be obvious from the name, positions text with respect to the centre of the display. Among the input validation commands is FETCH, which provides control over the number and type of characters which will be accepted as input from the keyboard. Curiously, although Commodore seem to be encouraging programmers to improve the structures of their programs (vis-a-vis the inadequate RENUMBER command) the ON KEY and CGOTO statements appear to be in direct contrast to this. For example, the ON KEY statement allows conditional branches to prescribed lines in response to input from the keyboard, while the CGOTO is an advanced branching facility in which the execution path of a program is determined, mathematically — something which is guaranteed to make the 'structured' bods wince! However, the ON KEY statement is quite special, in that it makes use of the interrupt signal generated when a key is pressed, to permit program branching without specific reference from a line. This means that any line, command, or statement could be interrupted, in whatever state of execution it may be in, and the pro-

gram will automatically branch to a preset line! This is 'old-hat' to machine code programmers, of course, but the only other machine that supports this type of facility from BASIC is the much-vaunted but elusive, Enterprise Microcomputer.

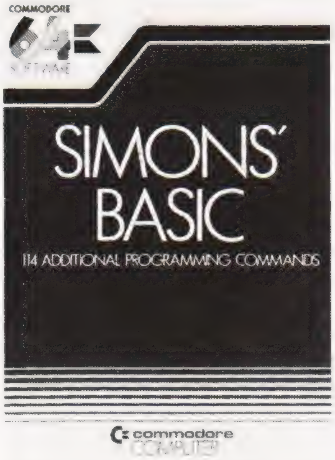
● **Arithmetic and Discs** Although the 64 already has its fair share of general arithmetic functions, a few of the supplementary commands which SIMONS' makes available to the programmer are quite noteworthy. One such example is FRAC, which returns the mantissa of a floating point number, e.g. FRAC(180/57.3) gives the number 14136126 (which is, of course, the fractional part of PI).

As always, it is a pleasure to note that SIMONS' allows numbers to be represented in the common bases, Hex, Denary and Binary and that the EXOR (exclusive OR) operator has also been included.

The disc commands have been improved considerably and although the changes are quite general, the improved and much abridged method of accessing disc-files is guaranteed to please, as is the DIRectory command.

● **Sprites and Graphics** Many of the greatest improvements which have been made by SIMONS', and indeed the other extensions, are in the areas of sprite-manipulation and general, hi-res graphics. SIMONS' has a multitude of commands which go well beyond providing the simplest line-drawing facilities. Single commands exist for drawing rectangles, specifying multi-colour graphics, filling areas bounded by an existing line-drawing (although this feature was found to be relatively slow) and for rapidly creating 'blocks' of colour. The DRAW command allows a complete picture to be created in a single command. If desired, the picture can then be rotated about the X and Y axes using the ROT command. Many other commands are present which also greatly simplify the specification and selection of background and foreground colours and other screen attributes, such as whether a colour is to be displayed 'flashing', and so on.

Sprites, or MOBs (Moveable Object Blocks) as they are referred to in the SIMONS' manual, are relatively easy to



design (relative to POKEing the necessary information) although as will be revealed, the other packages demonstrate that sprite-design and control can be performed in a more satisfactory way by using commands which are simpler both to use and understand. SIMONS' MOB commands were a little convoluted. Each MOB has first to be initialised and the machine instructed which block of MOB data is to be associated with a given MOB. This is quite complicated in practice and the same goes for SIMONS' MOB collision-detection apparatus.

● **SIMONS' Sound** Some of the strongest features of SIMONS' BASIC, are its sound commands. Full control is available over the Attack, Decay, Sustain and Release characteristics of an envelope, using the ENVELOPE statement and a similar level of dexterity is possible when specifying waveforms, using the WAVE command. Once the envelope, filter and voicing characteristics have been specified, large strings of notes can be described — but not played — using the MUSIC statement. This requires two parameters; the number representing the duration of each beat and a character-string containing note, octave, and time-measure information. This music can eventually be played using the PLAY command, which itself has a parameter, used to describe whether program execution is to continue as the music is played, or whether the program is to halt until the composition has terminated. If all of this sounds a trifle involved, this is simply because it is! However, it does allow precise definition of the sound and overall, the sound commands are very powerful, apart from being extremely well documented in the relevant chapter of the SIMONS' manual.

BC BASIC

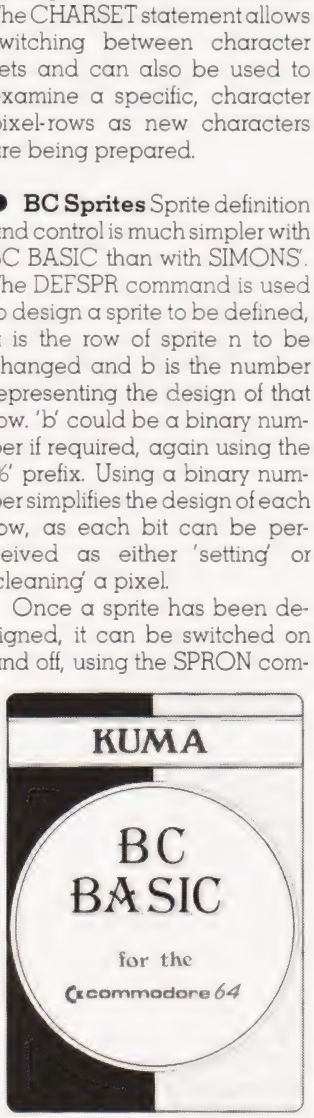
Probably the strongest rival to SIMONS' BASIC is BC BASIC, from Kuma Computers Ltd. This package contains many similar commands to those implemented in SIMONS', yet nearly all possess important syntactical differences, which somehow make it nicer to use. To begin with, however, let us note some of the commands available from BC BASIC.

A number of relatively basic graphics commands have been incorporated and amongst these is the BORDER command. This sets the screen-display border to one of the sixteen available with the 64. The PAPER command defines the main display area background colour, while INK selects the colour for the foreground.

The DEFUSR command allows the character-set to be redefined and the original character set can be returned by issuing the COPY statement. The CHARSET statement allows switching between character sets and can also be used to examine a specific, character pixel-rows as new characters are being prepared.

● **BC Sprites** Sprite definition and control is much simpler with BC BASIC than with SIMONS'. The DEFSPR command is used to design a sprite to be defined, a is the row of sprite n to be changed and b is the number representing the design of that row. 'b' could be a binary number if required, again using the '%' prefix. Using a binary number simplifies the design of each row, as each bit can be perceived as either 'setting' or 'cleaning' a pixel.

Once a sprite has been designed, it can be switched on and off, using the SPRON command and any sprite can be put into motion using SPRPOS n,X, Y, where n is the number of the sprite to be moved and X,Y are the coordinates to which it is to be moved. Dead simple.



One interesting thing to emerge from reading the BC BASIC manual is that displaying sprites affects the operation of the CPU, slowing it down in fact, and this apparently affects the timing of the cassette and disc operations — something which none of the other packages contain references to at all.

A command which I had never seen before was the SCRWAIT command. This instructs the micro to wait until the next TV blanking period before updating the display. This is designed to eliminate screen-flicker, which can happen if very fast screen operations are occuring and/or if the display contains a large number of objects in motion. This is an unusual feature to have available from BASIC.

In a similar way to the other extension packages, BC sports a full range of so-called 'structured' commands and statements, although nowadays, this is almost considered 'de rigueur'.

SOFTCHIP EXTENSIONS

Each company seems to have a different approach to selecting which commands are to be included in their language extension ROM cartridges. The designer of SIMONS' BASIC, for example, surveyed the variations to BASIC offered by other micros and even some minis but such a survey can only reflect a very general consensus of opinion and cannot cater for individual requirements. However, Whitby Computers' SOFTCHIP cartridges CAN be tailored to suit individual needs. This is made possible by placing into the cartridge only those commands specified by the purchaser. Naturally, it isn't quite as simple as it first appears — the purchaser must select his commands from a table supplied by Whitby. This table contains three columns: the first column shows the commands which can be implemented; the second, the number of bytes consumed by that command and the third, a brief description of the action of the command. The purchaser can select his commands to fill a 7100 byte area, so by carefully

noting the space required by each command, a list is compiled, representing those BASIC extensions of greatest benefit to that individual. The table contains a total of 106 different commands, covering virtually all of those areas covered by SIMONS' and BC. The only area which has not been touched is the 64's sound facilities, although I gather that the table is added to periodically and no doubt sound commands will be added.

Incidentally, I was fascinated to discover that Whitby had included a CORRUPT command, dubiously specified as being 'a negative-utility function for problem enrichment'!

PROGRAMMER'S FRIEND

If the prospect of designing one's own cartridge seems rather daunting, one can take advantage of two packages which contain related commands from the standard SOFTCHIP selection. The first of these is the Programmer's Friend. This cartridge contains a number of programming and debugging aids and a selection of additional disc-related commands, such as APPEND, which allows additional data to be added to an existing, sequential file. The CONCAT command links (ie concatenates) sequential files, whilst the RENAME and CATALOG or DIRECTORY commands perform their respective, as-should-be-obvious functions. Automatic line numbering has, understandably, been added and fortunately, possesses a default option of ten-line incrementing — unlike SIMONS'. Similarly, the RENUMBER command renumbers all GOTOs and GOSUBs, also unlike SIMONS'.

Few packages even admit that the Commodore 64 has an IEEE 488 interface but the 'Friend has a command, TIMEOUT, which (in contrast to rumours suggesting that it gives a printout of the week's London events!) sets the IEEE's timeout condition.

The HELP command is one of the greatest assets of this package. Giving the HELP command causes a line in which a recent error occurred to be printed out — a real time-saver. All Softchip commands can be instantly extracted from a program with the KILL command and the number of ACTUAL lines in a program can be calculated with the LINES function.

Access to the Commodore's resident machine code monitor can be obtained using the MON command, while REPLACE, a handy little command, allows the programmer to replace any string in a program with a new string. This allows variable identifiers to be altered at will, without resorting to 'hand-editing' the program.

SHRINK removes all unnecessary spaces and REMs from within a program. The qualifier 'unnecessary' is used to emphasise that some REMS definitely are necessary, as a few, admittedly poor, programmers allow program-branches to lines which only contain REMarks. Hence, all lines which begin with the REM statement are not removed.

The USER command is simply, fascinating. It is the only time that I have ever seen a BASIC command that allows new BASIC commands to be specified from BASIC. All the user is required to supply is the machine code for the routine and the BASIC interpreter can be forced to accept a new BASIC keyword, using USER, which adds a new, BASIC token and address to the Commodore's keyboard look-up table.

VAR is the equivalent of SIMONS' DUMP command, which prints out the identifiers and associated values for the variables used within a resident program.

This virtually completes the list of 'peculiar' commands available with the Programmer's Friend, although a complete list is available from Whitby, upon application.

SOFTCHIP BUSINESS COMMANDS

The second Softchip package from Whitby is the Business Commands package. Essentially, this package has been designed to add a range of commands which would commonly be constructed from a string of ordinary BASIC commands. An example of the special requirements of business programs would be an enhanced, keyboard input/output facility which could reject a number of specified keys. Such a command, as implemented in the

Business Commands package, is the DATIN command, designed to be a 'foolproof' keyboard-data entry command. Another similar command is the BLANK command which tests to see if a string consists of spaces, shifted spaces or a null, as often happens when just the Return key is entered in response to the INPUT statement.

Quite significant improvements have been made to the conditional control structures of the 64. CIF (Conditional IF) CEND (Condition END) ELIF and ELSE statements have been added. Basically, these commands allow conditions to be spread over several lines, ie the condition need not appear on the same program line as the CIF statement. This is a great improvement over the conventional IF THEN statement and its use has generally been restricted to Pascal and its derivitives until now.

Tailored error-handling routines can also be implemented with the aid of the ERROR command. This permits the execution of a program to continue from a chosen line, in the event of certain, 'trappable' errors, overriding the Commodore 64's existing error-handling facility.

This should give the flavour of the commands available from the Business Commands package, although, like the Programmer's Friend, complete details of all available commands are available from Whitby.

TO CONCLUDE

Naturally, the 'raison d'etre' of any language extension package is that it should dramatically reduce the amount of code, and hence effort, required to produce a desired effect from the machine. All of the packages discussed previously achieve this aim but each perform very differently in practice. SIMONS' BASIC possesses by far the greatest number of commands and the best quality manual of all but its commands can often be very 'long-winded' and were, in general, a little too indirect for my liking. Its price of £50 could also be a little off-putting.

Kuma's BC BASIC was nice and its commands posessed a directness which I liked very much — not too many parameters and an intelligent choice of command names. Again, as with SIMONS' a price of £57.50 may have a deterrent effect.

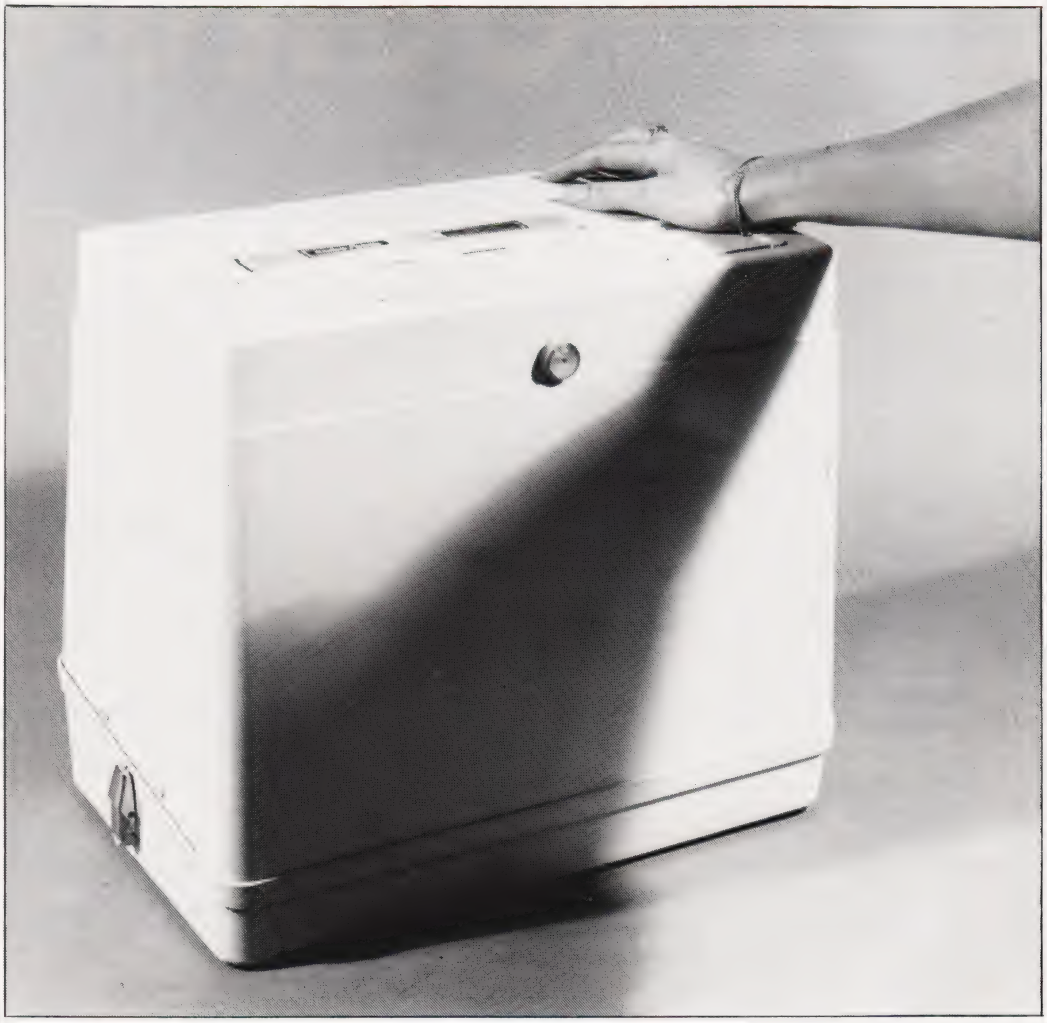
But my vote goes to the SOFTCHIP 64 range and its choice of commands. The ability to select which commands to incorporate into one's micro and indeed, a command which allows the programmer to add his own keywords, are very eye-catching features. Although one obviously does not get the range and sheer number of commands available from either SIMONS' or BC BASIC, to my mind, at £37.50, the purchase of a SOFTCHIP cartridge would be a very sound investment. Speaking of 'sound', however, do remember that at the time of writing, no specific sound-manipulation commands are available for SOFTCHIP cartridges.

Commodore Business Machines (UK) Ltd,
675 Ajax Avenue,
Trading Estate,
Slough,
Berkshire SL1 4BG

Kuma Computers Ltd,
12 Horseshoe Park,
Horseshoe Road,
Pangbourne,
Berkshire RG8 7JW
Tel: 07357 4335

Whitby Computers Ltd,
8 Chubb Hill Road,
Whitby,
North Yorkshire,
YO21 1JU
Tel: 0947 604966

TANDY TAKEAWAY



Simon Dismore

When Osborne launched their first portable the world's collective jaw dropped. But is portability just a marketing ploy? It depends on how well you do it — and Tandy have done it very well indeed with their new Z80A-based micro.

There is something magic about the idea of a machine that travels with you. A feeling of quiet pride, perhaps, when the neat box beside you on the train is revealed to be (gasp) a computer. But all too often the taste of adventure has soured with experience of unreliability and incompatibility. Will Tandy's new model 4P succeed where others have failed?

First, what exactly *is* a portable? The IBM PC, with all its separate boxes connected together with cables, is not. The battery-power CMOS machines like the Tandy 100 and the NEC 8201 certainly are — but they don't include CRTs and disc drives. The Tandy 4P, like the Osbornes before it, creates a category of its own.

This category has a new name — ***trans***portable. This means that you get a complete system in one casing, together with a carrying handle. It does ***not*** mean that you can run it off four AA cells wherever you may find yourself. No portable battery can keep up with the demands of disc drives and cathode ray tube for long enough to do any serious work, so you have to carry your machine from mains socket to mains socket in order to use it.

SEW, WHAT'S THE POINT?

Indeed, you probably wouldn't want to carry the Tandy around all the time in any case. At about 25 pounds, roughly twice the weight and size of a full briefcase, it is an uncomfortable strain to carry for any distance — from the office to the car presents no difficulty, but a trip through the rush-hour underground would be something to avoid. Compared with the original Osborne, the shape is much squarer. The reason for this is the generous 9" screen, but the first impression is of bruises to the knee when it jolts against you, and comments like "Is that a sewing machine?" do nothing to enhance the nonchalant air you contrive as you haul it into the pub.

One might be forgiven for rejecting 'transportability' as a marketing gimmick — what use is a portable machine which you can only use indoors? For use in a taxi or train, you need a genuine portable which runs off batteries, and communicates to you through a liquid crystal display. This type of machine is genuinely useful for making quick notes of meetings, keeping track of a short diary and address book, and simple data processing applications like stock counting and preparing quotations. But you will need patience and good eyesight to do any heavy work without a proper screen and disc storage.

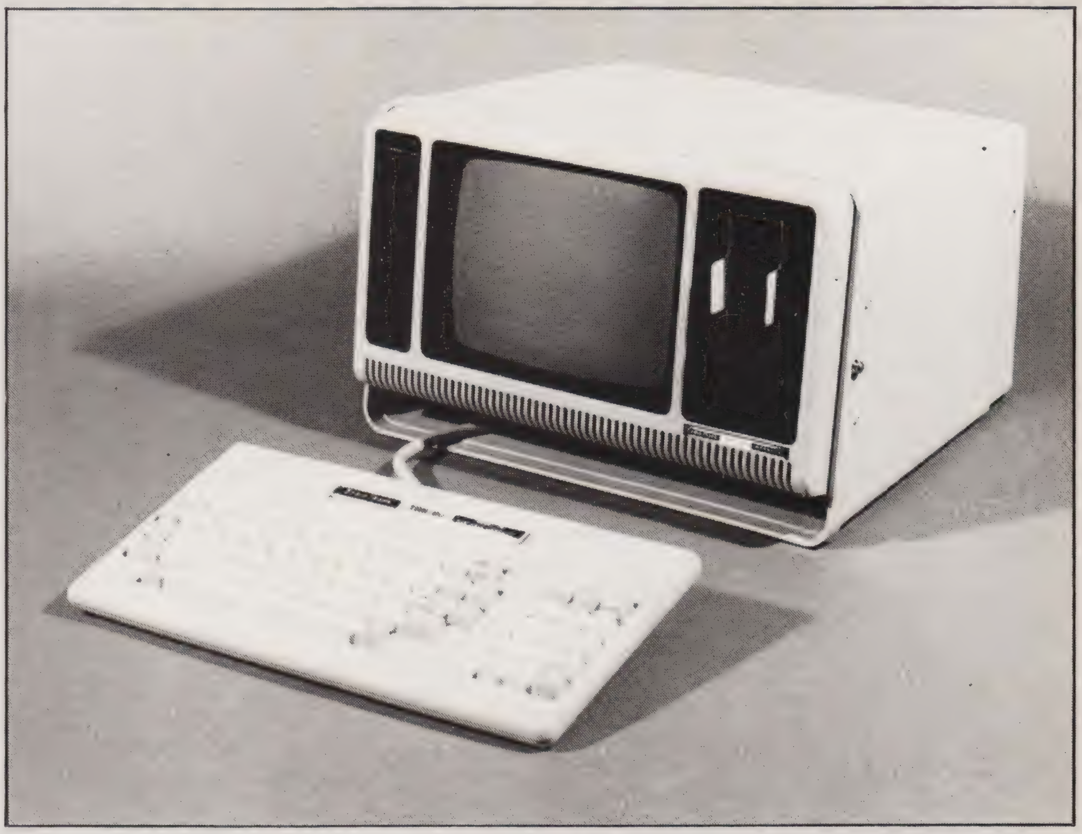
TRANSPORTS OF DELIGHT

This is where the transportables score heavily. You can pack your machine up in 30 seconds, go home, to an hotel or another office and be back at work with the minimum of delay and aggravation. This just isn't possible with a conventional three-box micro: the very thought of disconnecting and packing the components kills all your good intentions stone dead. So the Tandy scores as a machine for people who travel in their work (particularly if they travel by car) and should be a real hit with the self-employed, who are often forced to do all their (unprofitable) adminis-

trative tasks in unsocial hours.

There have been no compromises made to achieve transportability: with its rock-steady 9" green phosphor screen, dual 184K drives and tiltable keyboard, the Tandy looks every inch a business machine.

The business-like approach continues as you sit down for your first "hands-on" experience. Plug in and switch on, and a friendly message appears — "The Floppy Disc Drive Is Not Ready". Fair enough, put a disc in this time. Another message appears — "Close The Floppy Drive Door And Try Again". Each message appears in French and German as well as English, so you can note that the "Disketten-Laufwerk Nicht Bereit" if it makes you feel happier. The work that Tandy have put into friendly diagnostic messages is admirable — the ROM-based routines will catch every common error, and several that took some ingenuity to create. Any fool can learn to use the system in a short space of time (It shows! — Ed).



MAKE A DATE

Once the door is closed, the TRSDOS (pronounced "Triss-doss") 6.0 operating system signs on with a request for the date, which is unfortunately required in US format (MM/DD/YY). There is much to be said for the unambiguous British Standards suggestion of an international YY/MM/DD format, but the US will no doubt take some time to acknowledge this wisdom. Enter the date, and TRSDOS returns with "TRSDOS Ready".

This marks the first difference between TRSDOS and the more familiar CP/M operating system. For example, "A>" is CP/M's way of telling you that you are using drive A (drive zero). Unless you explicitly refer to another drive (eg: "B:"), CP/M expects to find whatever program you request on the current drive — so you need this constant reminder of which drive you are using. TRSDOS is too clever for this — the current drive is searched first, followed by all the other drives on the system until your program can be found. Of course, this means that you must be careful not to create two different programs with the same name, but this minor limitation is well worth the ease-of-use which TRSDOS offers.

CP/M will no doubt remain the single most popular operating system in the eight-bit market — the weight of existing users and software ensure its continuing success. Indeed, CP/M Plus is available from Tandy for the Model 4P, giving access to over 15,000 tried and tested products which are falling in price as the 16-bit world takes over. The only snag is that TRSDOS is in many ways a better operating system.

USING TRSDOS

Files are protected through a comprehensive scheme of disc labels, disc passwords, file owner and user passwords and no less than seven access attributes (execute, read, update, write, rename, remove and full access). Files are date stamped for creation and updating, as in Concurrent CP/M, and the directory and backup utilities will accept date ranges for selective display and duplication of files. All these options are invoked by entering explicit parameters with the commands (for example, BACKUP :0 TO :1 (DATE="06/01/84-06/30/84", QUERY=YES) will back-up all of June's files from drive :0 to drive :1, prompting for confirmation before each filename). This ensures that TRSDOS commands are easy to learn, though somewhat verbose in use.

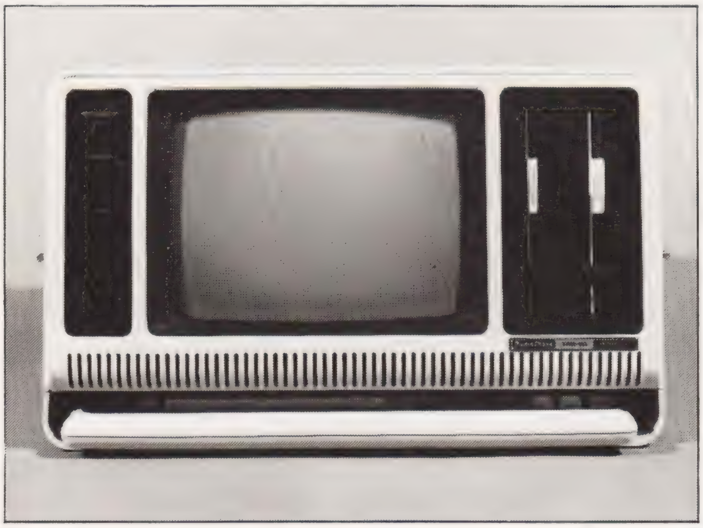
TRSDOS adopts the same concepts of logical and physical devices as CP/M (eg: under CP/M, the logical printer "LST:" might be either the physical lineprinter "LPT:" (which is usually attached to a parallel printer interface) or, say, a physical daisywheel printer "UL1:" on a serial interface). However, TRSDOS takes these facilities a great deal further, using an almost UNIX-like pipelining through devices. A logical device can be re-routed to another device or a filename — a directory listing can be put into a disc file, for example, for subsequent editing or display. Each device is controlled by a driver program and filter program. The driver program is responsible for the details of input/output to the device (handshaking, error correction, special control sequences, etc) while the filter program performs the more mundane task of converting input bytes into output bytes. This means that applications programs can become truly device-independent.

Imagine that you want to perform word processing in a foreign language, but your word processor will only accept the conventional ASCII character set. You can attach a filter to the screen (device *DO — "Display Output") which converts ordinary ASCII characters into special characters from the display ROM (eg: "[" might be converted to "[symbol for è]"). Your word processing software can now display European accents. An additional filter or device control program might also be required to ensure that your printer (*PR) can produce the appropriate character shapes. Remember, you don't need to make any changes to the WP software to achieve this — you merely change the behaviour of the logical devices on the system.

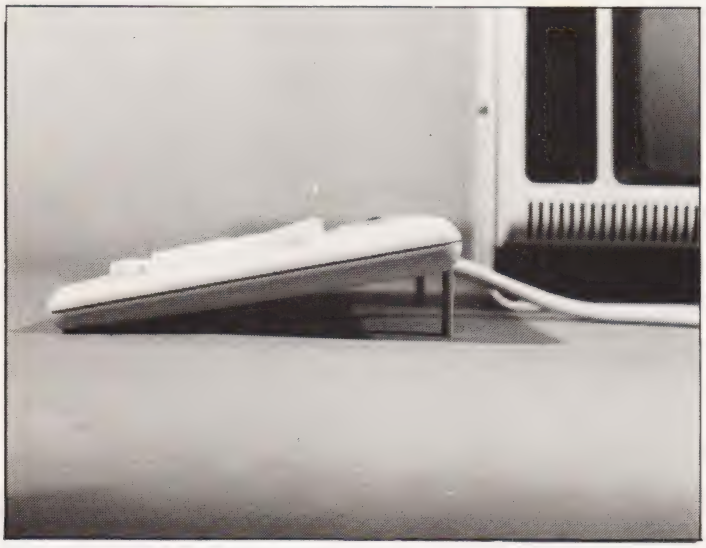
SPOOLING FOR SPEED

Any device can be buffered using the spool feature. Imagine that you are editing a fairly large BASIC program and want a printed listing. BASIC does not support 'background printing', so you would normally expect to twiddle your thumbs while your slow printer responds to LLIST. Under TRSDOS, you can fool BASIC into thinking it has a very fast printer. SPOOL lets you assign part of memory as a printer buffer (for very long documents, buffer space on disc can also be assigned). All you need to do is assign the spool buffer to the printer, return to BASIC, and LLIST takes seconds instead of 20 minutes. The operating system then sends your listing from the buffer to the printer 'in the background' while you continue editing

You get both a massive manual, and the handy pocket-size version (the former is not very transportable!).



During transportation the keyboard is stowed in a compartment under the main console . . .



. . . and during use, fold-down legs tilt it up at the rear for ease of use.

ATTRIB *filespec (parameters)*
Changes the protection of a file.
USER = *"password"* sets user password
OWNER = *"password"* sets owner password
PROT = *level* sets protection level for user.
EXEC
Execute only
READ
Read and execute
UPDATE
Update, read, execute
WRITE
Write, update, read, execute
RENAME
Rename, write, update, read, execute
REMOVE
Remove, rename, write, update, read, execute (total access except changing attributes with ATTRIB)
FULL
Total access
VIS specifies *filespec* as visible
INV specifies *filespec* as invisible

ATTRIB [:*drive*] (*disk parameters)*
Changes the protection of files on a drive.
LOCK sets user and owner passwords of unprotected visible files to disk master password
UNLOCK removes user and owner passwords of visible files whose passwords match disk master password
MPW=*"password"* states disk's current master password
NAME[="disk name"] specifies new disk name
PW[="password"] sets new disk master password

AUTO [*parameters*] [*]
[*command line*]
Stores a command line for automatic execution each time TRSDOS starts up. (AUTO by itself deletes the current AUTO command line.)

* disables the **BREAK** key during boot; disables **ENTER** override of AUTO *parameters:*

:drive specifies drive to store AUTO command line on
?[:*drive*] displays AUTO command stored on *drive*
=[:*drive*] executes AUTO command stored on *drive*

BACKUP [*partspec*] [:*source drive*]
[TO] [:*destination drive*] [(*parameters*)]
Duplicates all or some of the files on a diskette.
MPW=*"password"* states source disk master password
SYS backs up system and visible files
INV backs up invisible and visible files
MOD backs up files modified since last backup
QUERY=YES prompts for each file
OLD backs up only files that already exist on destination disk
NEW backs up only files that do not already exist on destination disk
X backs up with no system disk in Drive 0
DATE=*"M1/D1/Y1-M2/D2/Y2"*
backs up files modified on or between two dates
=*"M1/D1/Y1"* backs up files modified on date
=*"-M1/D1/Y1"* backs up files modified on or before date.
=*"M1/D1/Y1:"* backs up files modified on or after date

DIR [*partspec*] [:*drive*]
[(*parameters*)]
Lists the directory of a drive or file.
ALL displays all directory information for specified files
INV displays non-system invisible and visible files
MOD displays files modified since last backup
NON enables non-stop display mode
PRT prints directory on printer
SYS displays system and visible files
DATE displays files with today's date
DATE=*"M1/D1/Y1-M2/D2/Y2"*
displays files modified on or between the two dates

A SELECTION OF THE MORE NOTABLE TRSDOS COMMANDS

="*M1/D1/Y1*" displays files modified on date
="*-M1/D1/Y1*" displays files modified on or before date
="*M1/D1/Y1-*" displays files modified on or after date
SORT=NO prevents alphabetical sorting of directory entries

FORMS [(*parameters*)]
Sets up printer options.

DEFAULT returns all options to their start-up values
ADDLF issues linefeed after every carriage return
CHARS=*number* sets number of characters per printed line
FFHARD issues a form feed (Top of Form) character instead of a series of linefeeds
INDENT=*number* sets number of spaces to indent lines that are longer than CHARS
LINES=*number* sets number of lines on each page
MARGIN=*number* sets left margin
PAGE=*number* sets physical page length in lines
QUERY prompts for each parameter
TAB specifies that tab characters are to be translated into the appropriate number of spaces
XLATE=X'aabb' specifies a one-character translation
aa is the character to be translated
bb is the character aa is translated to

LIST *filespec* [(*parameters*)]
Lists contents of a file to the display or printer

ASCII8 displays graphic and special characters along with text
NUM numbers lines in ASCII text files
HEX specifies hexadecimal output format
TAB=*number* specifies that tab stops are to be number of spaces apart for ASCII text files (default=8)
PRT directs output to printer
LINE=*number* sets starting line number
REC=*number* sets starting record number
LRL=*number* formats output using specified logical record length when in hex mode

PATCH *filespec* (*patch commands*)
Changes the contents of a disk file

address=value specifies patch by memory load location. Changes contents of memory beginning at address to value.
Drecord,byte=value specifies direct modify patch. *record* specifies record to be changed (in hex); byte specifies first byte to be changed (in hex.)
Frecord,byte=value checks location specified by the D *patch command* to make sure it currently contains the data specified by *Frecord, byte.* Also used with REMOVE parameter (see below) to remove a patch and replace it with the original data.
Lcode specifies library mode patch. *code* is binary coded location in the format *nn* where change begins.
value is a series of hex bytes in the format *nn nn nn*. . ., or a string of ASCII characters in the format "string".

PATCH *filespec1* USING *filespec2* [(parameters)]
Makes changes contained in *filespec2* to *filespec1.*

YANK removes the patch specified by *filespec2* from *filespec1.* *filespec2* contains code in the *address* format.
REMOVE removes the patch specified by *filespec2* from *filespec1. filespec2* contains code in the *Drecord, byte* format.

PURGE [*partspec*]: *drive* [(*parameters*)]
Deletes files.

QUERY=NO removes files without prompting
MPW=*"password"* states disk master password
INV removes invisible and visible files
SYS removes system and visible files
DATE=*"M1/D1/Y1-M2/D2/Y2"* removes files modified on or between two dates
="*M1/D1/Y1*" removes files modified on date
="*-M1/D1/Y1*" removes files modified on or before date
="*M1/D1/Y1-*" removes files modified on or after date

ROUTE *devspec1* [TO] *devspec2*
ROUTE *devspec1* [TO] *filespec* [(REWIND)]
ROUTE *devspec1* (NIL)
Routes a device to another device, to a disk file, or to nothing (NIL).

SPOOL *devspec* [TO] *[filespec] (parameters)*
Establishes an output buffer for a device
NO turns off spooler and resets *devspec*
MEM=*number* specifies amount of memory buffer (in K) for spooler
BANK=*number* selects one of three 32K memory banks to use as spool buffer (0, 1, or 2)
DISK=*number* specifies amount of disk buffer (in K) for spooler
PAUSE temporarily suspends output to *devspec*
RESUME restarts output to *devspec* after a PAUSE
CLEAR clears the spool buffer

SYSTEM *(parameters)*
Selects certain options of your TRSDOS system. In the following SYSTEM commands, *switch* is YES or NO.

SYSTEM (ALIVE[=*switch*])
Displays a moving character when task processor is running.
SYSTEM (BLINK=*switch*)
SYSTEM (BLINK=*number*)
SYSTEM (BLINK[,*parameter*])
Control the cursor character. *number* specifies ASCII value in decimal for cursor character
parameter is LARGE or SMALL. If *parameter* is omitted, the cursor returns to its start-up character and size.
SYSTEM (BREAK[=*switch*])
Enables or disables Break key.
SYSTEM (BSTEP=*number*)
Sets default bootstrap step rate used with FORMAT utility.
SYSTEM (DATE[=*switch*])
Turns on or off the start-up date prompt.
SYSTEM (DRIVE=*drive, parameters*)
Sets the following parameters for *drive:*
CYL=*number* sets default number of cylinders used with FORMAT utility (35 to 96)
DELAY=NO/YES sets DELAY time for floppy disks
DISABLE removes access to *drive*
ENABLE allows access to drive that has been disabled
STEP=*number* sets step rate for *drive*
DRIVER=*"filespec"* configures hard drive
WP[=*switch*] sets software write protect
SYSTEM(FAST)
Switches system to fast clock rate.
SYSTEM (GRAPHIC)
Specifies that printer can reproduce TRS-80 graphic characters during screen prints
SYSTEM (RESTORE[=*switch*])
Determines whether all drives are to be restored when the system is reset.
SYSTEM (SLOW)
Switches system to slow clock rate.
SYSTEM (SYSRES=*number*)
Adds TRSDOS system overlays into high memory. *number* is 1-5 or 9-12.
SYSTEM (SYSTEM=*drive*)
Assigns drive as system drive.
SYSTEM (TIME[=*switch*])
Turns on or off the start-up time prompt.
SYSTEM (TRACE[=*switch*])
Continuously displays contents of Program Counter.
SYSTEM (TYPE[=*switch*])
Turns on or off the type-ahead feature.

and debugging.

For the beginner, some of these utilities might seem hard to understand. Indeed, Tandy's (very comprehensive) documentation declares that a number of the utilities are for 'advanced programmers'. For a programmer, the more advanced utilities are a godsend. Patches to programs can be made automatically from prerecorded lists of bytes, instructions can be set-up for automatic execution whenever the system is booted, and an extremely powerful 'Job Control Language' is available for stringing your programs together. In addition, a powerful facility of TRSDOS's otherwise standard Microsoft BASIC is the ability to execute the utilities from within a BASIC program by adding a command string to the SYSTEM function. This very useful feature could be adapted to create simple housekeeping routines within a BASIC program suite, or even to write an entire user-friendly front end environment. The extensive features for password, access and date-stamped protection mean that it should be quite impossible for a user to do anything that the programmer did not intend.

As an operating system, TRSDOS appears to be ideal for developing highly secure custom applications — making the model 4P a very attractive candidate for point-of-sale terminals, stock control systems and perhaps small military systems.

FACTSHEET	Tandy Model 4P
CPU	Z80A
Clock	4 MHz
RAM	64K
Language	Microsoft BASIC on disc
Weight	11 kg
Size	16½" by 13¼" by 9¾"
Display	Text 80 or 40 columns by 24 lines (64 columns by 16 lines in Model III mode) Low res block graphics
I/O	RS232C interface Tandy parallel interface Hard disc interface Integral dual 184K 5¼" disc drives
OS	TRSDOS 6.0 CP/M (optional)
Options	Expansion memory, 64K RAM bank switched — £99.95 including VAT 5M Winchester hard disc (external unit) with intelligent backup utility and hard disc operating system — £1999 including VAT. Extra expansion hard discs (up to three additional 5M units) — £1699 including VAT. High resolution graphics (640 by 240 pixel monochrome resolution, plus graphics BASIC with subroutine library) — £199.95 including VAT.

HARDWARE

The hardware itself is businesslike, if not particularly exciting. The keyboard offers a standard typewriter layout, though some of the standard ASCII characters are only available using combinations of keys. Backspace (CHR$(8)) and tab (CHR$(9)) are generated by the cursor left and right keys, which might confuse some word processing software, and escape (CHR$(27)) can only be produced using Shift together with cursor up. This is a shame, as the Escape key is very useful for switching between modes in software — BASIC uses it, for example, to toggle out of insert mode when editing a program line — but given the restricted space on the keyboard it is an understandable compromise. Less acceptable is the positioning of the Break key at exactly the position where the backspace character is normally found, which can have infuriating results if you are not used to it. A configuration utility can be used to control the break key function — essential for those secure applications. There is a minimal numeric keypad, with 0-9, decimal point and Enter, and three programmable function keys, though the documentation gave no information about setting the function keys from BASIC.

The screen is a fast but otherwise quite ordinary 9" monochrome tube, displaying 80 or 40 columns by 24 lines (or 64 by 16 lines when emulating the older TRS-80 model III) with low resolution graphics using the extended ROM character set. A high resolution graphics option has been announced, which offers 640 by 240 pixels resolution plus a graphics BASIC with a subroutine library and assembly language interface. This would be adequate for normal scientific work, but is scarcely going to break Hewlett-Packard's traditional domination of the laboratory market. Brightness and contrast can be controlled separately, so the screen is legible even under quite severe glare conditions.

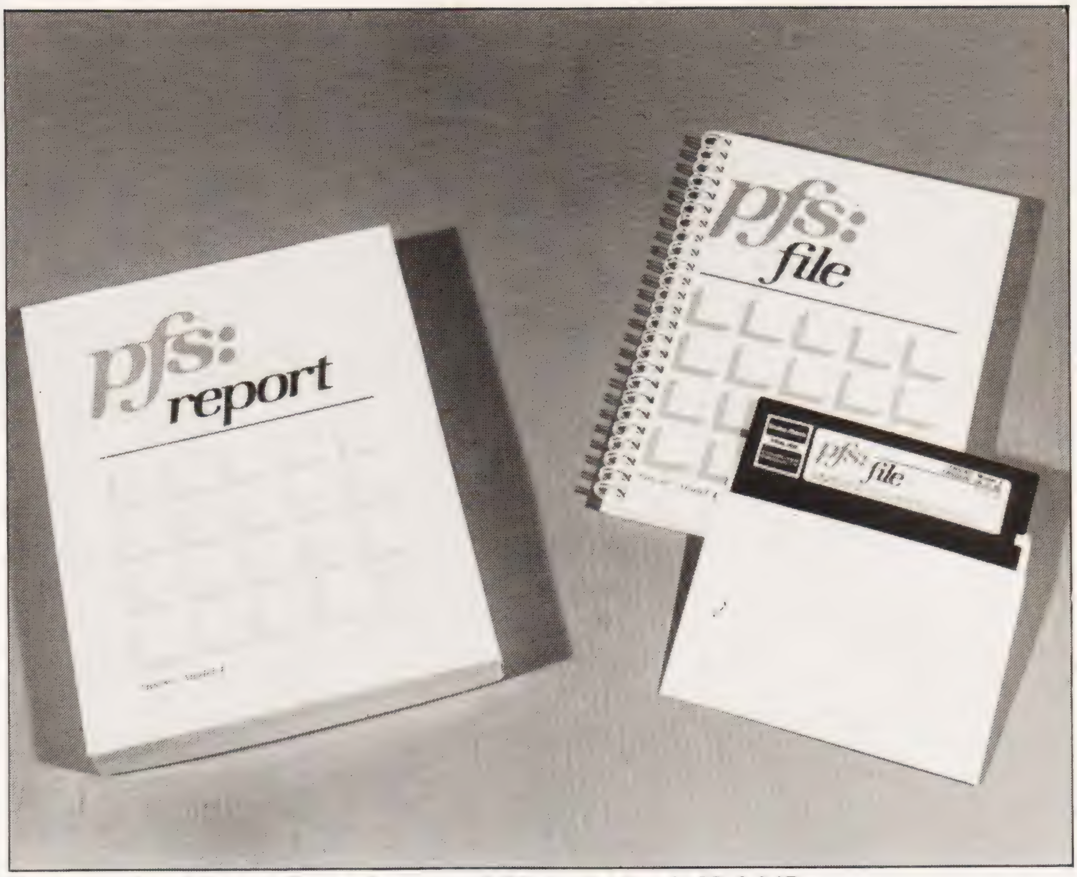
The twin double density 5¼" drives offer 184K storage and promise a data transfer rate of 250 KBits/second. They didn't seem that fast in normal use — but they were certainly reliable enough. Disc formatting and verification utilities were fast and easy to use, and the supplied BACKUP utility makes light work of selective disc copying with TRSDOS's usual attention to passwords and date stamping. There is also a 5 MB hard disc option, though as this introduces a separate box to an otherwise integrated system it rather reduces the transportability of the machine, and at £1999 (including VAT) it is scarcely a bargain.

INS AND OUTS

A standard RS-232C interface (25-pin D connector) is provided, for asynchronous communications up to 9600 bps, together with communications and character conversion software. There is also an edge connector for a TTL parallel printer cable, which could no doubt be converted by a competent dealer for the standard 36-way Centronics interface.

The minimum 64K RAM can be expanded to 128K for an additional £99.95. The Z80A

can only directly address 64K, so bank switching is used to access the extra memory, which means descending to the darker areas of assembly language. The spooler can use up to 32K of the extra RAM, which would yield a generous printer buffer — ample for most applications except word processing long reports (which should in any case be split into smaller files for security).



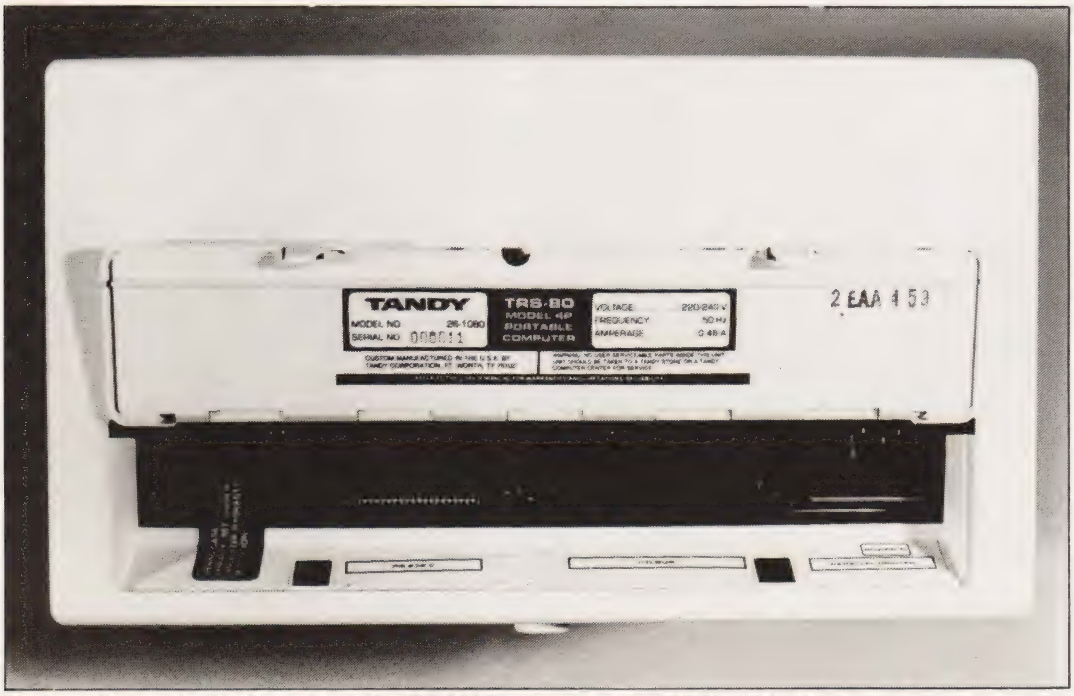
Two examples of the excellent software available for the Tandy Model 4P.

SOFT SELL

With the review machine came copies of PFS:file and PFS:report, each priced at £69.95. They provide easy-to-use data management using full screen editing. Unlike more traditional products (the much-maligned dBase II, for example) PFS:FILE requires absolutely no programming skill or knowledge of data types, and can be learned in under an hour by anyone who already knows the machine.

PFS:file is used to define screen formats (rather like the Autocode product for dBase II) each format having up to 100 pages, and each page having up to 32 fields for data entry and retrieval. As this means a theoretical maximum of 3,200 fields per record we were not able to test the software to its limits, but found it an astonishingly simple product to use for simple lists, diaries etc, producing output to screen or printer in a variety of easily defined formats.

Searches using PFS:file were reasonably fast using the primary key (the first field of the first page), though rather slower on other fields. All activities — seraches, deletions, print formats and so on — use the same screen design that is used for entering data, so applications development is a trivial task. Each field is terminated by using a cursor key — so the Return character can be used to create multi-line entries, still something of a rarity on most data management software. Individual words and phrases within long entries can be searched on, using a generous selection of wildcard characters and logical operators. This means that relatively sophisticated enquiries (like 'find all meetings I've booked with Mr Brown') can be made on essentially unstructured entries — ideal for the businessman who has neither the time nor the inclination to make exact entries in the database in the way that other products often require. PFS:report can then be used on PFS:file databases to generate listings with totals and subtotals, columns headings, and so on. Both the packages have excellent documentation and represent excellent value for money.



Under a flap at the back, we have the RS-232C connector, the I/O bus (blanked off here), the parallel printer port and the mains socket.

CONCLUSIONS

The lasting impression given by the Model 4P was of an efficient, reliable machine with above average documentation and utilities. It is by no means a 'state-of-the-art' product, but a solid workhorse which can run a vast range of tried and tested software under the TRSDOS, LDOS and CP/M operating systems. At £1499 (including VAT) it represents reasonable value for money if you need an easily transported system. Tandy themselves are a stable, mature manufacturer who will not disappear overnight, and deserve the success that this product will no doubt bring.

THE (BIO)MECHANICS OF COMPUTING

Richard Nicholls

The application of computers in sport is big business in America with vast sums being invested in the pursuit of athletic perfection. The application of biomechanics to locomotive illnesses has already proved invaluable in both Britain and America — could it be a case of "We have the technology . . ."

Americans own more recreational hardware, planes, boats, off-road vehicles or campers, than any other nationality. Ownership of a private aircraft is not especially unusual in the USA, some people have their own executive jets. Quite a few own larger jets, Boeing 707 or similar — but not many of them are in the process of building a runway in their back garden, especially not a runway big enough to accommodate a jumbo jet they've just added to their fleet. Clearly this person is a millionaire. In fact he's making a million dollars a day or something equally ridiculous, selling his keep fit invention.

You may be forgiven for thinking that anyone who can earn that much money must have invented the most up-to-date piece of high-tech keep-fittery ever but in fact you'd be wrong. That's not to say that the Nautilus exercise machine isn't truly amazing but it's hardly high tech. These days, high tech in keep-fit and its associated fields means only one thing — computers.

For the most part the latest computerised keep-fit gear is all part and parcel of various institutional researches or development programmes. Hardly surprising, really, since it costs millions to develop and operate and only large bodies have the necessary

financial muscle to provide funds for the investigation of microchip muscle.

However, there is at least one person who earns enough money to have developed and used a computer program geared to the needs of only one individual. Not a financier, pop star or drug-smuggler, the money comes from sport. Tennis, to be precise, at which the top-earning professional is one Martina Navratilova. There are around 50 big tennis tournaments in the season: the really big money comes at Wimbledon and the US, French and Australian Open Championships. Total prize money available to women competitors at last year's Wimbledon was $614,000. Ladies' Singles Champion (Martina Navratilova 1982 and 83) prize money last year was $96,000. Ladies' Doubles winners (guess who) split $37,000. The US Open (no prizes as to who won that last year) pays about the same, others slightly less and then a descending scale.

SMART LADY

Anyone who wins regularly can expect to earn a great deal. They will also need to re-invest a great deal in order to stay ahead, which is precisely what Miss Navratilova has done. Once a week, wherever she is in the world, she receives a highly confidential report, written in a code known only to one other person. This document, the 'Smartina Report', is the output from a computer. The program was devised by Robert Haas and contains all kinds of dietary and exercise information, all of it specifically related to Martina and no-one else. It is designed to increase her endurance and stamina and has done exactly that for its individual beneficiary. Miss Navratilova is not, however, the only tennis player whose details are recorded in the computer memory. There is also a great deal of information about those people against whom she may reasonably expect to play, such as Chris Evert-Lloyd. This is also included in the weekly Smartina report when relevant and Martina uses it to study form when preparing for a match.

What she doesn't do — or at least hasn't admitted — is use the computer to improve her own style and performance. That is reasonably understandable, since that would increase the cost of the installation to a point where not even her staggering success rate could produce enough prize money to cover it. But it is being done and on both sides of the Atlantic too, although it will soon be apparent that as far as this kind of research goes, money doesn't talk so much as drawl, with a noticeable American accent.

1984 is an Olympic year, of course, and the Games are being held in the city which is best qualified for the title of millionaires' head office. Chief hangout for the obscenely rich, Los Angeles has been preparing for a financial boom which will make its already huge daily spending rate pale into relative insigificance. It's not only Los Angeles which is in preparation, however, for it seems that scarcely has one Olympiad ended then competitors begin preparation for the next. It's true also that this preparation appears to be most necessary, as year after year athletes striving for greater and greater achievements and perfection push the human body further and further. So if you want to be in among the medals, you have to get training and stay there, timing it so that the peak of fitness is reached at the right moment.

DOING IT IN STYLE

But in most, if not all, field sports, stamina and endurance are not the only things which count. Style is also vastly important, the more so in gymnastics, since style counts towards final marks, on top of the fact that a clean style can actually assist the gymnast to jump higher, further, longer. All those involved are aware of this and it is at this point that the coach makes his major input, watching the athletes in action and advising them on ways to improve their performance — lift that leg higher, tuck it in tighter, earlier or faster. Or at least, the coach advises the athlete on ways which he *believes* will improve performance. But now the guesswork is being taken out of it.

At the Olympic Training Centre in Colorado Springs, the athletes who will be representing the USA this June are undergoing their usual stringent training sessions, only this year there's a difference. Under the guidance of Dr Charles 'Chuck' Dillman, the new science of Biomechanics is being enlisted to give some extra help. Biomechanics as far as the athletes are concerned, involves being plugged into computers so that peformance can be analysed and ways for improvement suggested and tried.

SENSORS ON

Input for this comes from several sources; electrodes fitted to the athletes themselves, high-speed film or video of field performance and also from pressure sensors on the floor or the apparatus in use. Film, and lately video, has been a normal training tool in many sports for years but in this case the film is used to convert the competitor's actions into stick men whose actions can then be endlessly re-run, analysed and re-modelled in the computer. Combined with the measurements taken from pressure pads this can, in theory, help the athletes improve their performance.

The same sort of thing exists in this country, although not on the same scale or to the same degree of sophistication as the Colorado Springs setup. Over here the activity is limited to a few athletics hopefuls, nominated by the BAA, who are computerised by the Department of Physical Education at Leeds University. High-speed film produces the stick men, and the resulting 3D graphic display is available to athletics coaches. Onto that display is added the centre of gravity of the body in question — the computer then offers 'suggestions' as to where that centre of gravity ***should*** be at any given moment and how high off the ground it should be. What it does not do is inform athletes how to achieve the ideal; it merely points out what that optimum is and leaves it to athlete and coach to work out how to reach it.

In any case, the information need not always be that accurate. In any sport where style, speed and strength combine against gravity and mass the computer can help. But not very much . . .

Take gymnastics. Height off the mat counts but how to achieve it? During a complicated series of exercises acceleration is vital and positioning of limbs critical, both to achieve good landings and strong takeoffs, as well as smooth flight through the air. In order to advise correctly on this the computer needs to know the mass of the body in question, as well as of each individual limb. However, these things are all joined together and cannot be successfully weighed individually without the aid of a saw, after which the whole point of the excercise would be lost!

For that reason, they are all based on an assumed average, using quoted data from medical textbooks (which was originally obtained from cadavers). The computer assumes body density to be 1, the same as water and uses the information from the textbook to apportion mass according to height and weight. It can allow for the fact that some people have short bodies and long legs, or vice versa, but it changes these things on its stick-men graphic display based on fractional parts of total bodyweight, meaning that its assumed mass for individual limbs is almost certainly wrong. All its future data concerning the positioning of limbs in flight is therefore suspect.

FUNDING

This is one area in which other advancing sciences may soon be able to help: X-rays and ultrasonics, for example, should be able to assist with information about bone size, structure and weight and much work in some areas is being carried out quite independently of biomechanics. The only obstacle at present is cooperation and collation. Even so it means more money needs to be spent and it's money which is at present lacking in this country. In any case, as far as sports achievements are concerned, it may not actually be worth it, since it may only produce a tiny improvement — two or three percent — in total accuracy.

What may perhaps be reasonably cheaply achieved — although at the cost of maybe two years development of software — is a realistic computer model of what is possible for the human body, so that the computer may be best employed providing goals, setting targets for athletes to aspire to. This is the real crux of the matter; all these various things are possible with computers, but just how much use are they? So far only Martina Navratilova seems to have turned a computer to practical sporting advantage and that's not truly in the field of biomechanics. The graphic displays may be useful in other spheres, such as dance training and choreography but that has yet to achieve fruition. The whole problem of useful applications was covered in a BBC Schools series although **The Computer and the Gymnast** was only one of a series which examined application rather than theory.

More valuable, in fact, is the application of biomechanics to illnesses affecting locomotion like arthritis, even to post-operative care after fitting of artificial joints and limbs. In this country a great deal of work is being done in this area by the Bio-Engineering department of Strathclyde University but again, money is the major obstacle. Britain lags years behind the USA in these two related fields of bio-mechancis and bio-engineering; the Leeds system was in prototype form eight years ago, was usuable but not affordable four years ago and is only now coming properly on stream. In the USA, at least one system has been commercially available for some time.

As far as post-operative care is concerned, the Americans are again miles in the lead, thanks mostly to the work of Dr Jerrold Petrofsky. His work with paraplegics has largely concentrated on using a computer to replace the brain in cases where the spinal cord is severed or crushed. The computer sends electrical commands directly to the muscles needed to walk; it even contains a built-in program to provide balance, the software taking thousands of hours and the final program running to no less than 3391 lines. Petrofsky's big success was 23-year-old Nan Davis who, confined to a wheelchair at the age of 20 after a car crash, walked up to collect her diploma at graduation, using Petrofsky's computer, which had now been miniaturised to fit into her handbag.

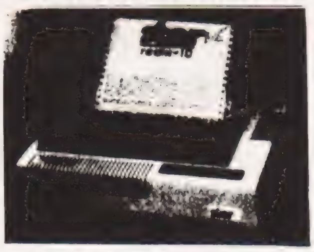
The future is even more exciting, since it is planned to reduce the computer to single-chip status and implant it like a heart pacemaker, with fine wires running under the skin directly to each individual muscle. The money for this has been provided by the American government since Nan Davis brought Petrofsky into the limelight and it is these millions of Federal dollars which helped PC Phillip Olds, paralysed since a gunman shot him down three years ago, to take his first hesitant steps very recently, joining five others who have risen to their feet with the aid of Dr Petrofsky and his team.

By 1990 Petrofsky hopes that refined versions of his 'pacemaker', powered by a tiny watch battery, will allow paraplegics to walk forwards and backwards, turn, bend and climb stairs, which means that money spent now on research into bio-mechanics and bio-engineering will never be wasted. It seems a cruel irony that at present most of the money comes from the Federal Government in America but in this country is provided by universities, the BAA and in one instance, from a national newspaper. The Daily Mail (in conjunction with the Metropolitan Police) is looking into ways of funding a co-operative experiment in British hospitals to aid paraplegics using Dr Petrofsky's proven equipment.

This year, when the Americans clean up on gold medals at the Olympic Games, you may rest assured that it will not be steroids or other drugs but a computer that will have made the difference. And it's good to know that the millions it will have cost to develop the software isn't simply to be squandered on sport.

No. 4 June/July 1984
Digital&Micro
ELECTRONICS
An Argus Specialist Publication
£1.95
Strip Chart Data Plotter For Experimenters
HARDWARE REVIEWS
The Microprofessor 1+
Data Directors and Routers
Data Recorder for Home Computers
Under The Micro-scope
INSPECTING THE ATMOS
Programmable Slide Projector Interface For Commodore 64, BBC Model B or Vic 20
Continuing... THE DIGITAL OSCILLOSCOPE
Concluding... ROBOT ARM CONVERSION
OUT NOW!

DOT MATRIX PRINTERS



ALL PRINTERS HAVE CENTRONIC PARALLEL INTERFACE UNLESS OTHERWISE STATED. ALL PRINTERS HAVE HI-RES. DOT ADDRESSABLE GRAPHIC MODE. PLEASE SEND SAE FOR FULL DETAILS.

EPSON		
FX 80 160 CPS 10" wide fric & pin feed	£344 + VAT	£395
FX 100 160 CPS 15" wide fric & tract feed	£431 + VAT	£495
RX80 F T 100 CPS 10" wide frict & tract feed	£261 + VAT	£299
RX80T 100 CPS 10" wide tractor feed	£234 + VAT	£269
LQ1500 200 CPS Letter quality printer		
8143 RS 23 Interface for FX and RX printers	£29 + VAT	£33
8148 RS232 Interface with 2K buffer xon xoff	£60 + VAT	£69
Ribbon Cartridge for RX80 FX80 & MX80	£7 + VAT	£7
Ribbon Cartridge for FX100 & MX100	£12 + VAT	£13
STAR		
Gemini 10X 120 CPS 10" wide fric & tract feed	£200 + VAT	£229
Gemini 15X 120 CPS 15" wide frict & tract	£295 + VAT	£339
Delta 10 160 CPS 10" wide frict & tract feed	£327 + VAT	£375
Delta 15 160 CPS 15" wide frict & tract feed	£414 + VAT	£475
Radix-10 200 CPS NLQ 16K Buffer Par & serial interface downloadable characters incredible value at	£431 + VAT	£495
Radix-15 15" wide version of Radix-10	£518 + VAT	£595
SEIKOSHA		
GP100A 50 CPS 10" wide tractor feed	£156 + VAT	£179
GP250X 50 CPS 10" wide trac feed ser & par	£196 + VAT	£225
GP100VC 30CPS 10" wide VIC 20 & 64 interf	£174 + VAT	£199
GP500A 50 CPS 10" wide tractor feed	£187 + VAT	£215
GP550A 50 CPS 10" wide friction & tract correspondence quality	£234 + VAT	£269
GP100AS Serial interface ideal for Spectrum with microdrive	£174 + VAT	£199
SHINWA		
CP80 80 CPS 10" wide friction & tract feed	£187 + VAT	£215
Spare ribbon cartridge for CP80	£6 + VAT	£6
RITEMAN		
Compact 120 CPS 80 col frict & tract feed	£200 + VAT	£229
CANON		
PW1080A 160 CPS 10" wide 27 CPS NLQ 24 x 16 matrix	£283 + VAT	£325
PW1156A 160 CPS 15" wide 27 CPS NLQ 24 x 16 matrix	£347 + VAT	£399
COLOUR PRINTERS		
Seikosha GP700A 7 colour 50 CPS printer	347 + VAT	£399
Canon PJ1080A 7 colour 40 CPS ink jet printer	£391 + VAT	£449
CPP40 40 column 4 colour battery operated	£118 + VAT	£135

ALL OUR PRINTERS HAVE 1 YEAR WARRANTY

DAISYWHEEL PRINTERS

JUKI 6100		
20 CPS Bi-Directional Logic seeking 10 12 15 CP1 + PS Spacing 2K buffer bes selling Daisywheel	£334 + VAT	£395
Singer sheet feeder unit	£217 + VAT	£249
Tractor Unit	£95 + VAT	£109
RS 232 Interface	£52 + VAT	£59
Spare Daisywheel	£17 + VAT	£19
BROTHER HR-15		
13 CPS Bi directional 10 12 15 CP1 + PS	£344 + VAT	£395
Keyboard Unit	£139 + VAT	£159
Single Sheet Feeder Unit	£217 + VAT	£249
Tractor Unit	£95 + VAT	£109
DAISY STEP 2000		
20 CPS Unidirectional 10 12 15 CP1	£260 + VAT	£299
SMITH CORONA TP1		
12 CPS 10 12 CP1 unidirectional	£208 + VAT	£239

PROFESSIONAL MONITORS

- *SANYO
- *MICROVITEC
- *FIDELITY
- *PHOENIX

SANYO		
DM8112 12" Green Screen 18 MHZ Hi-Res	£86 + VAT	£99
DM2112 12" Green Screen 15 MHZ Monitor	£66 + VAT	£75
CD3117 14" RGB Normal Res Colour Monitor	£173 + VAT	£199
CD3115 14" RGB Medium Res Colour Monitor	£286 + VAT	£329
CD3125 14" RGB High Res Colour Monitor	£391 + VAT	£449
MICORVITEC CUB		
1431 MS 14 RGB Normal Res Colour Monitor	£173 + VAT	£199
1451 MS 14 RGB Medium Res Colour Monitor	£321 + VAT	£369
1441 MS 14 RGB High Res Colour Monitor	£417 + VAT	£479
FIDELITY Colour Monitor		
CM14 14 RGB RGBY Composite sound col	£187 + VAT	£215
PHOENIX		
Phoenix High Res 12" Green Monitor	£66 + VAT	£75
Phoenix High Res 12" Amber Monitor	£69 + VAT	£79

BBC MICROCOMPUTER SYSTEM

WE ARE AN OFFICIAL BBC COMPUTER DISTRIBUTOR

DEALER ENQUIRIES ARE WELCOMED

Acorn Electron £199 INC VAT

WE SUPPLY FREE 30 HOUR BASIC BOOK AND A DUST COVER WITH EACH BBC COMPUTER

BBC is the best microcomputer currently on the market 32K RAM 32K ROM 8 modes of operation full colour full size keyboard internal expansions such as disc interface speech synthesizer Econet interface — in short it is a personal computer capable of expanding into a small business system

BBC Microcomputer Model B	£347 + VAT	£399
BBC Mod B + disc interface	£408 + VAT	£469
BBC Mod B + Econet interface	£389 + VAT	£447
BBC Mod B + disk and Econet interfaces	£450 + VAT	£517
BBC 100K disk drive	£230 + VAT	£264
BBC dual 800K disk drive	£699 + VAT	£803
Torch Z80 disk pack including Z80 2nd processor 64K RAM and CPN operating system + Free Perfect Software	£696 + VAT	£799
BBC Teletext receiver (Aug)	£196 + VAT	£225
BBC cassette recorder and lead	£26 + VAT	£29
Disk interface kit (free fitting)	£84 + VAT	£96
Mod A to Mod B upgrade kit	£70 + VAT	£80
Fitting charge for A to B upgrade kit	£20 + VAT	£23
16K memory upgrade kit	£30 + VAT	£34
Games paddles	£11 + VAT	£12
[illegible]	[illegible]	£102
[illegible]	[illegible]	£22
User guide	[illegible]	£10
Econet interface (free fitting)	[illegible]	£69
Speech interface (free fitting)	[illegible]	£54
BBC [illegible]	[illegible]	£34
Parallel printer cable	£10 + VAT	£11
BBC word processor (view)	£52 + VAT	£59
BBC Fourth language cassette	£15 + VAT	£17
BBC Lisp language cassette	£15 + VAT	£17

APPROVED ECONET SERVICE CENTRE

WE STOCK A LARGE RANGE OF SOFTWARE FOR BBC MICRO INCLUDING ACORNSOFT BBC SOFTWARE LONGMAN SOFTWARE PLEASE SEND LARGE STAMPED ADDRESSED ENVELOPE FOR FULL DETAILS

TORCH Z80 2nd PROCESSOR
For only £347 + VAT

Torch 2nd Processor Z80 is supplied with perfect writer (a powerful Word Processor) perfect speller (spelling checking program — I should have used one for [illegible] this advert!), Perfect Filer (A Database Program), Perfect Calc (Spread Sheet). It includes 64K memory (Almost 60K available to user) [illegible] BBC Computer

Z80 2nd Process + Perfect Software	£347 + VAT	£399
Z80 Processor + Perfect Software + Dual 800K Disk Drives	£695 + VAT	£799

GUARANTEED LOWEST PRICES

100% BBC COMPATIBLE MITSUBISHI AND TEAC SLIMLINE DISK DRIVES

These drives are supplied ready cased with all the necessary cables formatting program and User Guide. There are some very useful utilities included on formatting disc e.g.

- DISASSEMBLER This is 6502 machine code disassembler
- DUP To copy and rename a file on disc
- FORMAT Formating program for 40 & 80 tracks
- FREE This utility provides a disk usage analysis
- MDUMP Enables you to display and modify any part of BBC memory
- MERGE Merge a number of text files into one file
- RELOCATE Downloads a basic program to &E00
- SDUMP Screen dump for EPSON in all graphic modes
- VERIFY Verifies every sector on a disk
- MENU A flexible menu program

Single drive 100K 40 trks single sided	£147 + VAT	£169
Dual drive 200K 40 trks single sided	£286 + VAT	£329
Single drive 200K 40 trks double sided	£173 + VAT	£199
Dual drive 400K 40 trks double sided	£330 + VAT	£379
Single drive 400K 80 trks double sided	£199 + VAT	£229
Single drive 400K 40/80 trks switchable DS	£217 + VAT	£249
Dual drive 800K 80 trks double sided	£382 + VAT	£439
Dual drive 800K 40/80 trks switchable DS	£408 + VAT	£469

All above drives are low power slimline (0.3 A typ at + 12v and 0.4 A at + 5v per drive). Normally extra power supply is not required. The BBC Computer power supply is designed to drive two low power drive (IT IS NOT DESIGNED TO DRIVE INTERNAL ROM BOARD)

40 Track SS DD diskette (10 Box)	£18 + VAT	£20
40 Track DS DD diskette (10 Box)	£23 + VAT	£26
80 Track SS DD diskette (10 Box)	£28 + VAT	£32
80 Track DS DD diskette (10 Box)	£30 + VAT	£34

ALL ABOVE DISKETT ARE CERTIFIED EITHER MEMOREX OR DETALIFE

YOUR CONTACTS AT AKHTER ARE

DEALER BULK ENQUIRIES	HAMAYUN MUGHAL
TELEPHONE ORDERS	PAULA HAYES
DEALER ORDERS	TONY GLOVER
BUSINESS SYSTEMS ENQUIRIES	DENNIS SUTCH
EXPORT ENQUIRIES	MAHAMAD EDIB
ECONET SYSTEM TECHNICAL ENQUIRIES	ALAN LAFFOLEY
ACCOUNTS	CARON ANDREWS
DESPATCH	PAUL SWIFT
REPAIRS	JOHN MAULE

BUSINESS SYSTEMS

APRICOT — SIRIUS — SANYO
IBM — TEXAS — TORCH

APRICOT Portable Executive Computer 16 Bit Micro 256K RAM up to 1.44 megabytes floppy disk storage 3½" Sony disks Portable brief case styling Modem with auto dialler (optional) hard disk optional Vast software library (compatible with Sirius 1)
Apricot with Single Drive and Monitor — £1690 + VAT
Apricot with Double Drive and Monitor and Free Printer or 2nd Monitor — £1890 + VAT

SANYO PROFESSIONAL COMPUTER
16 Bit Micro 128K RAM expandable to 256K Single or Double Disk Drive built in full colour graphics (640 × 200 pixels in 8 colours) IBM compatible Free software Sanyo MBC 550 128K RAM single drive and free software including Wordstar and Calcstar — £749 + VAT
Sanyo MBC555 128K double drive and free software including Wordstar Calcstar Infostar Datastar etc — £999 + VAT

SIRIUS 1 Sirius 1 Computer with 128K RAM and 1.2 megabyte Floppy disk storage including CP M 86 MS DOS and Microsoft Basic — £2195 + VAT
Sirius 1 Computer with 256K RAM and 2.4 megabyte Floppy disk storage including C C M 86 MS DOS & Microsoft Basic — £2895 + VAT

TORCH
CF500 COMPUTER with colour monitor includes Twin 800K disks Tronline Operating System free perfect software (perfect writer perfect speller perfect calc perfect filer) Torchmail Plus (Electronic Mail) Torch Mass (Financial modelling with graphics) Torchtel (Viewdata Access System) Executive Aid (Menu System) Datel 600 (Link to outside mainframe services) [illegible] — £2950 + VAT

TEXAS INSTRUMENTS PROFESSIONAL COMPUTER
TI 502 Dual 320KB Drives 256K RAM Up to 10 megabyte on board storage possible TI speech command hardware option brings a new level of user friendliness to computing. Allows voice operated keyboard control (Optional) — Price £2650 + VAT

MAINTENANCE CONTRACTS ARE AVAILABLE

COMPLETE WORDPROCESSING SYSTEMS INCLUDING DAISYWHEEL PRINTER
From £695 + VAT

SYSTEM 1 BBC Micro model B Wordwise Phoenix monitor Smith Corona TP1 Daisywheel Printer Cassett Recorder + all the necessary cables and documentation — £696 + VAT **£799.25**

SYSTEM 2 BBC Micro model B + Disk Interface Phoenix Monitor 100K Single Disk Drive Wordwise Smith Corona Daisywheel Printer + all the necessary cables and documentation — Price only £896 + VAT **£1029.25**

SYSTEM 3 BBC Micro model B + Disk Interface Phoenix Monitor Dual Disk Drives 200K Wordwise Smith Corona Daisywheel Printer + all the necessary cables and documentation — Price £999 + VAT **£1148.85**

SYSTEM 4 BBC Micro model B + Disk Interface Sanyo High Res Green Monitor Wordwise (or view) Juki 6100 Daisywheel Printer Single 400K Disk Drive + all the necessary cables and documentation — £1049 + VAT **£1206.35**

SYSTEM 5 BBC Mod B Micro + Disk Interface Sanyo High Res Green Monitor Wordwise (or view) Juki 6100 Daisywheel Printer Dual Disk Drive 800K + all the necessary cables and documentation — Price £1195 + VAT **£1374.25**

SYSTEM 6 Sanyo 16 Bit IBM pc Compatible Micro 128K RAM Single Disk Drive 160K Wordstar Calc Star Sanyo High Res Green Monitor Juki 6100 Daisywheel Printer + all cable and documentation — Price Only £1145 + VAT **£1316.75**

SYSTEM 7 Sanyo 16 Bit IBM pc Compatible Micro 128K RAM Dual Disk Drives 360K Wordstar Calc Star Mailmerge Infostar Spellstar Datastar Sanyo High Res Green Monitor Juki 6100 Daisywheel Printer + and all necessary cables and documentation — Price £1395 + VAT **£1604.25**

SYSTEM 8 All the components of System 6 but with Sanyo High Res Colour Monitor and Wordstar with Colour Facilities — Price £1445 + VAT **£1661.75**

SYSTEM 9 All the components of System 7 but with Sanyo High Res Colour Monitor and Wordstar with Colour Facilities — Price £1695 + VAT **£1949.25**

WE CAN ALSO SUPPLY PACKAGES WITH COMPONENTS OF YOUR OWN CHOICE. PLEASE PHONE (0279) 412639 AND ASK FOR HAMAYUN MUGHAL FOR A PRICE.

AKHTER INSTRUMENTS LTD T A

AKHTER

COMPUTER GROUP

28/29 BURNT MILL
HARLOW, ESSEX CM20 2HU U.K.
Tel. HARLOW (0279) 443521 Telex 818894 AKHTER G

ORDERING INFORMATION:

We accept official orders from UK Government and Education establishments. Carriage is Free (UK only) for normal delivery. If express delivery is required please add £8.00 + VAT per parcel. We accpet telephone orders on Barclay and Access Card please ring (0279) 443521 (10 lines). All cheques made payable to AKHTER INSTRUMENTS

Access

VISA

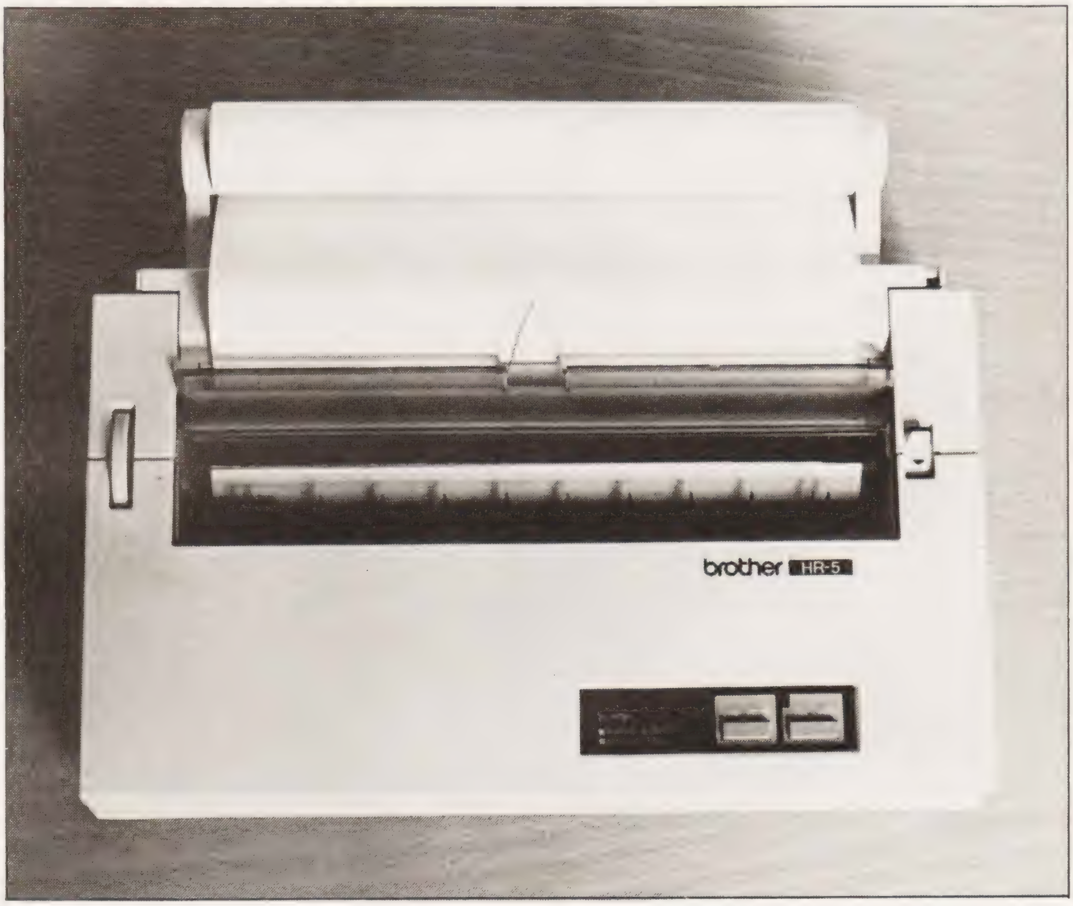
N.B. ALL VAT INCLUDED PRICES ARE THE CORRECT PRICES. EXCL VAT PRICES HAVE BEEN ROUNDED UP TO NEAREST POUND.

OPENING HOURS: MON-FRI 9am-6.30pm, SAT 10am-5pm
We welcome callers, no parking problems.

BABY BROTHER

Paul Gardner

If you're still looking for a printer for less than £200 that takes A4, will print 80 or more columns, has several different typestyles and hi-res graphics, you might have missed the quiet arrival of the new Brother HR5 Thermal Transfer Printer.



This machine follows in the footsteps of the popular EP (electronic portable) range of typewriter/printer/'terminals'. If I hadn't seen a pre-release model of the HR5 at the last Which Computer Show then I might not have known about it until it arrived in one of Birmingham's leading department stores; a refreshing change in these days of 'promises now, delivery later'.

The physical size of the printer is approximately 12" x 3" x 2½" and it comes packed in polystyrene in a large carrying box. The package includes one roll of A4 paper, a set of 4 HP2 batteries, three ribbons and an instruction manual. An optional mains adaptor was offered at £11.95.

IT'S A SET-UP

Setting up the printer was very straightforward, simply a matter of fitting one of the ribbons, loading the paper roll holder and connecting up the computer. The instructions in the manual are very clear on the correct procedures. I had, of course, read the manual from end to end before opening the box, so I knew how to start the print test (self-printing facility) which is detailed two pages from the back of the manual!

My first impressions were seriously marred when I found that it would not work with the mains adaptor supplied, but did work on batteries. Further enquiries established that the AC adaptor used with the EP range of printers is not powerful enough for the HR5. Brother should be supplying a suitable adaptor soon.

PRINT CHOICE

The HR5 uses a dot matrix thermal printing system, so it can be used with either thermal paper or plain paper with a cassette ribbon. It is extremely quiet in printing and the only significant noise is the winding on of the ribbon and the movement of the print head as it lifts off or presses onto the paper at the end of each line. The HR5 uses ninety-six ASCII characters and has sixty-three character generator graphics. Two typestyles are available — Pica (standard) across eighty columns, or Elite (selected from the computer) across ninety six columns. Characters can be printed enlarged or condensed and may be emphasised or underlined. There is even the facility to print "reduced double width enlarged characters"! Superscripts and subscripts are available as well as bit image (high resolution) printing.

Some confusion arises if you try to work out how many different styles you can have on the same print line, but a table is provided in the manual which attempts to clarify the situation. All of these print styles are available by sending control codes to the printer from the computer. Many of the available ESCAPE codes are Epson FX80 compatible. Control and escape sequen-

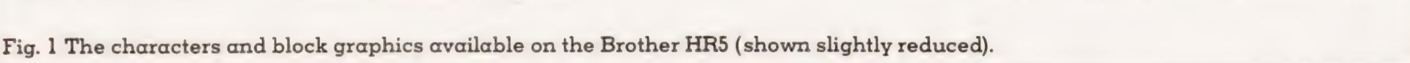


Fig. 1 The characters and block graphics available on the Brother HR5 (shown slightly reduced).

Fig. 2 Standard bit-image high-res graphics.

Fig. 3 Double density bit-image HRG.

Normal Pica characters
Reduced Pica characters
Enlarged Pica

Superscripts : Let $A=BX^{(C+D)}$
Subscripts : $FF_{(16)}=255_{(10)}$

Reduced Superscripts : Let $A=BX^{(C+D)}$
Reduced Subscripts : $FF_{(16)}=255_{(10)}$

Emphasised Pica characters
Emphasised Enlarged

Reduced double width Pica

Normal Elite characters
Enlarged Elite

Auto Underline is selectable
on all the above styles.

Fig. 4 These are all the available type styles.

ces allow you to use many of the usual features such as form feeds, tabulation, setting your own tab stops, page length, line pitch and so on. There is also fitted the usual collection of DIP switches to hardware select functions such as zero character font i.e. 0 or Ø, default line pitch 1/6" or 1/9", bell on/off and various sets of international characters. One peculiar feature is a DIP switch to control line feeds after carriage returns, which has the options of 'fixed' ie CR LF, or AUTOFEED XT which uses the state of one of the Centronics connector lines to determine LF. If your connector does not support this connection then the input seems to be held high, ie no LF after each CR.

If you have the RS232 interface fitted, the baud rates are switch selectable from 110 to 9600 with a factory set of 300. Bit lengths and parity are switch selectable.

SCREEN DUMPING

I use my HR5 with a Spectrum and a Kempston Centronics interface and it is possible to produce high resolution screen dumps if you load the interface program EPSONHR and make the following changes. Set DIP switch 1-2 to OFF on the HR5, and for the software, use POKE 23545, 201 to produce no LF after CR, and POKE 23450, 4 to produce the correct line spacing when in bit mode.

The diagrams show the proportions and densities of bit image printing available. The format of Fig. 3 ie double density bit image, is available for Spectrum/Kempston users using the EPSONHR program as above but also poking POKE 23452, 76.

PROS AND CONS

If all this gives you the impression that the HR5 is an inexpensive 'quality' printer, you may be wondering 'What's the catch?'. Well, the only drawbacks really are the print speed and running costs.

The HR5 runs at a full printing speed of 30 characters per second but the overall speed depends on other things. Usual printing is bi-directional and logic-seeking. Bit image, reduced, emphasised or reduced double width printing is, however, uni-directional and the printer takes as much time to do a CR as the line last printed. Printing is slowed down further still if you are using a ribbon and not thermal paper. At the end of each line of print, the amount of ribbon just used is wound into the cassette before the print head moves again. You'd have to see this in action to understand but it does mean that little ribbon is wasted which is an advantage.

SPEED V. QUALITY

As I've said, print speed is affected by the use of thermal or plain paper, and so is quality. Thermal paper is quicker but fainter whereas plain paper is slower but the quality is high; when printing emphasised characters it is almost up to letter standard.

The second drawback is cost. A4 thermal paper costs about 3p per sheet which works out at around 22 square feet per pound. This compares favourably with printers such as the Sinclair or Amber but is expensive compared with most plain paper printers. Ribbons cost £2.50 each and from my work so far could be expected to provide between thirty and forty pages of A4 which works out at between 6p and 8p per sheet. It is almost essential that you buy a mains adaptor. Although only rated at 6 Watts with a 6V supply, this draws 1 amp of current, which means that batteries don't last long!

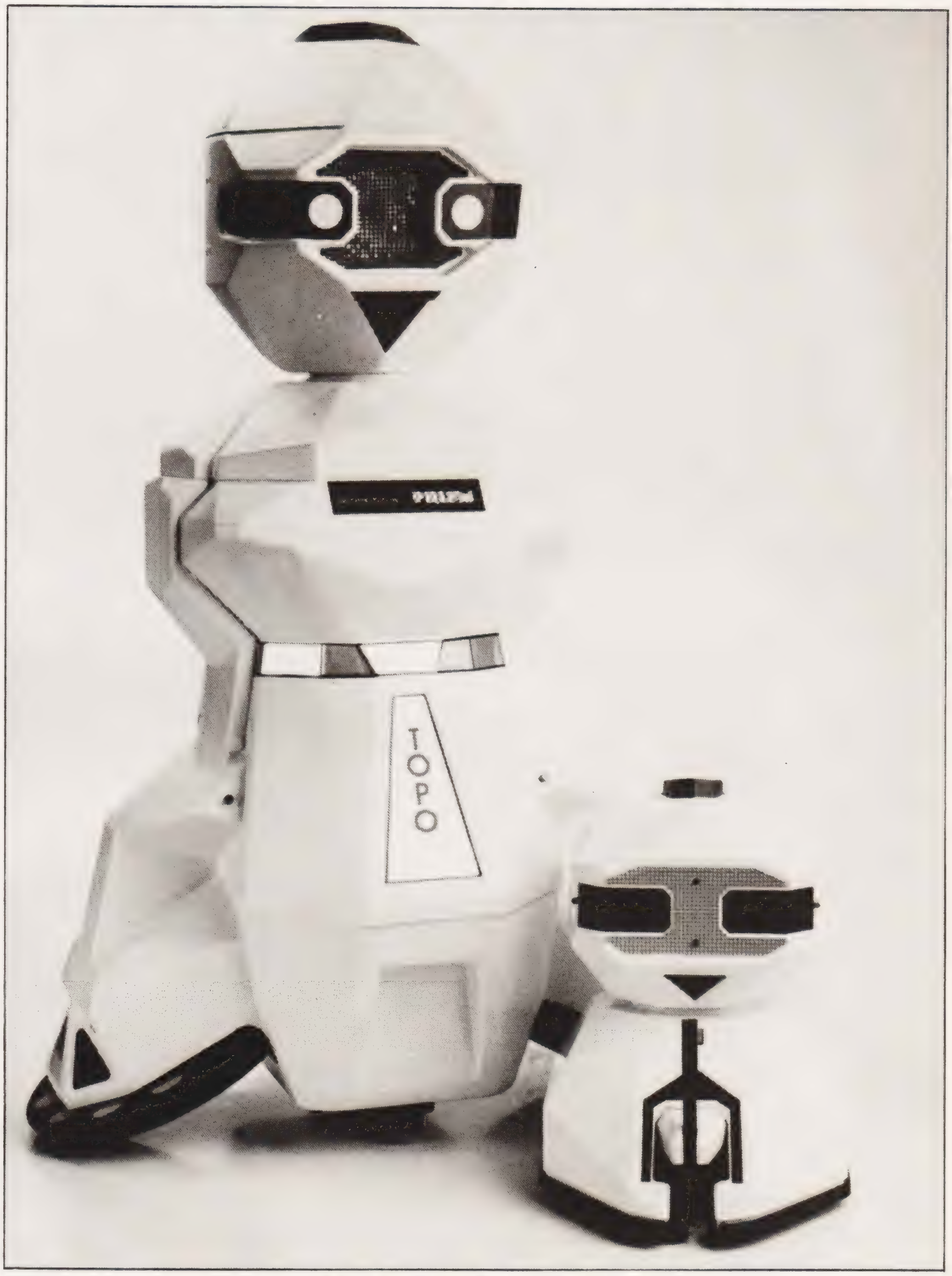
TO CONCLUDE

There is no doubt that the Brother HR5 is a high quality printer with facilities usually only available on printers over £300. Some people may be deterred by its slow speed and fairly expensive running costs but it is neat and, with battery power, portable and what is very important to many people, quieter than almost anything else. It seems to me to be the ideal first printer for the home computer enthusiast.

Price £179.95 (inc VAT)
Ribbons £2.50 each
Thermal paper £3.00 per 100 sheets

Optional mains adaptor (Brother B2) £11.95

MIND AND MOVEMENT



Liz Coley and Steve Colwill

Daleks have been with us since 1963 — the last twenty years have seen the emergence of many other equally celebrated automatons. But is it really possible to construct a robot capable of independent movement or is it just the wishful thinking of intergalactic movie directors?

In general, robots can move in two ways. Firstly, they may have a fixed geographical position but be able to twist, turn and extend limbs, as is the case with the robot arms currently used in industry. Secondly, robots may move around their environment. Of course these may be combined in a single machine with the ability to carry out tasks in several locations. Such a 'being' would be the basis of the popular concept of a robot. The techniques required to produce 'static' robots for a variety of tasks are fairly well advanced. on first glance it would appear that problems associated with changing location are relatively minor and indeed it is a straightforward task to design robots that can move within a specially constructed environment, such as a flat factory floor with few obstructions or along a set of rails. Difficulties start to arise when a robot needs to navigate its way through the human world of corridors, tables, stairs, and people!

MODEL BEHAVIOUR

In order to negotiate our environment, we rely on a constant stream of information from our senses. The data received is continually processed via the millions of neural pathways within the brain. This monitoring mostly occurs without us being conciously aware of it, although it is possible to become acutely aware of one particular input. Try concentrating on the feelings in your fingertips as they hold this magazine, or the point of maximum pressure exerted on your bottom by the chair you are sitting on! If you were deprived of any one sense, movement would become much more laborious — have you ever tried negotiating an unfamiliar flight of stairs in complete darkness? Although it might appear obvious it is much easier to tackle familiar stairs under the same circumstances because our minds contain a complex internal model of the external world which allows us to demote the task of navigation to the subconcious level. How many times have you followed a familiar route, intending to depart from it at some point, and suddenly found yourself

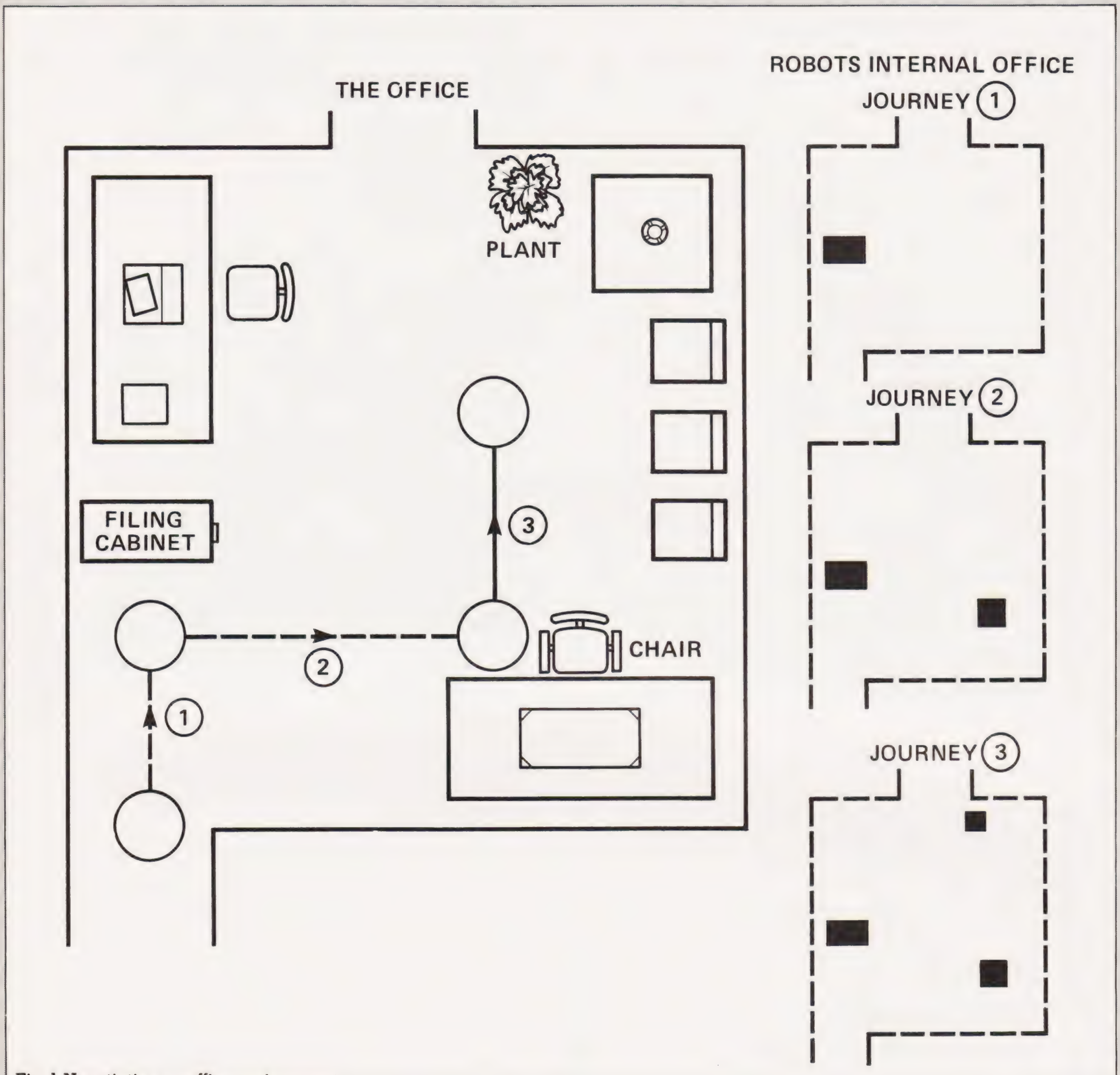


Fig. 1 Negotiating an office environment.

way off course having inadvertently followed the usual path?

If robots are to roam freely in our world they must have some form of sensory input from their surroundings and the ability to use this data to construct an internal model of the immediate environment.

If we look at the requirements for a wheeled robot capable of dealing with an office-type environment, a suitable design would need the following elements. Given that our robot has the capability to control forward motion and rotate about its axis, it still has to be able to find its way around, negotiating obstacles as it goes. Before the robot can make any meaningful decisions about its direction, it must have some way of obtaining information about its environment. There are several methods by which it can do this. 'Sight', in the accepted meaning of the word, would be desirable but visual methods of computer perception are still very limited. Instead, our robot could 'see' in the same way as a bat by bouncing ultrasonic waves off nearby objects and measuring the time taken for the echo to return. Ultrasonics are a convenient and accurate way for the robot to judge its relative position. The very high frequency sound waves emitted are harmless to living things (provided a suitable frequency is chosen) and as they are non-electromagnetic they do not interfere with electronics circuits, TV or radio. In addition, the wavelength of the high frequency signals is short in comparison to the distances being measured. It is this property that makes ultrasonic measurement over short distances so accurate.

Now the robot has been provided with the sensory input required to determine its relative position, it can be programmed to assimilate this data into its own internal model of its environment. As the robot moves around so pieces are added to the model making it a more accurate reflection of the real world as time goes on. Hence the robot can learn about its surroundings.

JOURNEY 1

The robot scans ahead and picks out the filing cabinet. It does not interpret this as such; only as an obstacle to be avoided.

JOURNEY 2

The robot is now moving in a different direction and picks out the chair. It can add this object to its internal model

which now consists of two items and their relative positions.

JOURNEY 3

As the robot continues its voyage of discovery it adds more detail, such as the pot plant by the door.

Once the machine has a fair idea of the office layout it can plan ahead, allowing more specific tasks to be carried out.

Of course, in the real world objects such as these do not always remain stationary. This raises another dilemma for our plastic pal. If it senses an object in a place where according to its internal model, there should not be one, which does it believe, its sensors or its internal model? Either could be wrong and more data is needed to decide which is correct.

To provide more data and increase safety, alternative sensors are required. These could be in in the form of bumper bars that stop the robot on immediate contact with any obstruction and wheel shaft encoders that provide additional information concerning distance travelled. These extra inputs add support to the two primary sources of positional information, the ultrasonics scanner and the internal model. If it is found that the internal model is incorrect then suitable modifications to the model can be made. The assimilation of these various data inputs and their incorporation into the controlling program increases the need for sophisticated onboard processing

DECISIONS, DECISIONS

Most current microcomputers process information serially, that is, one piece of data at a time in a stream. The requirement for our robot to deal with several inputs at once could be accomplished in this manner using a queueing principle but this would probably preclude movement at a sensible pace. Our robot would spend most of its time decid-

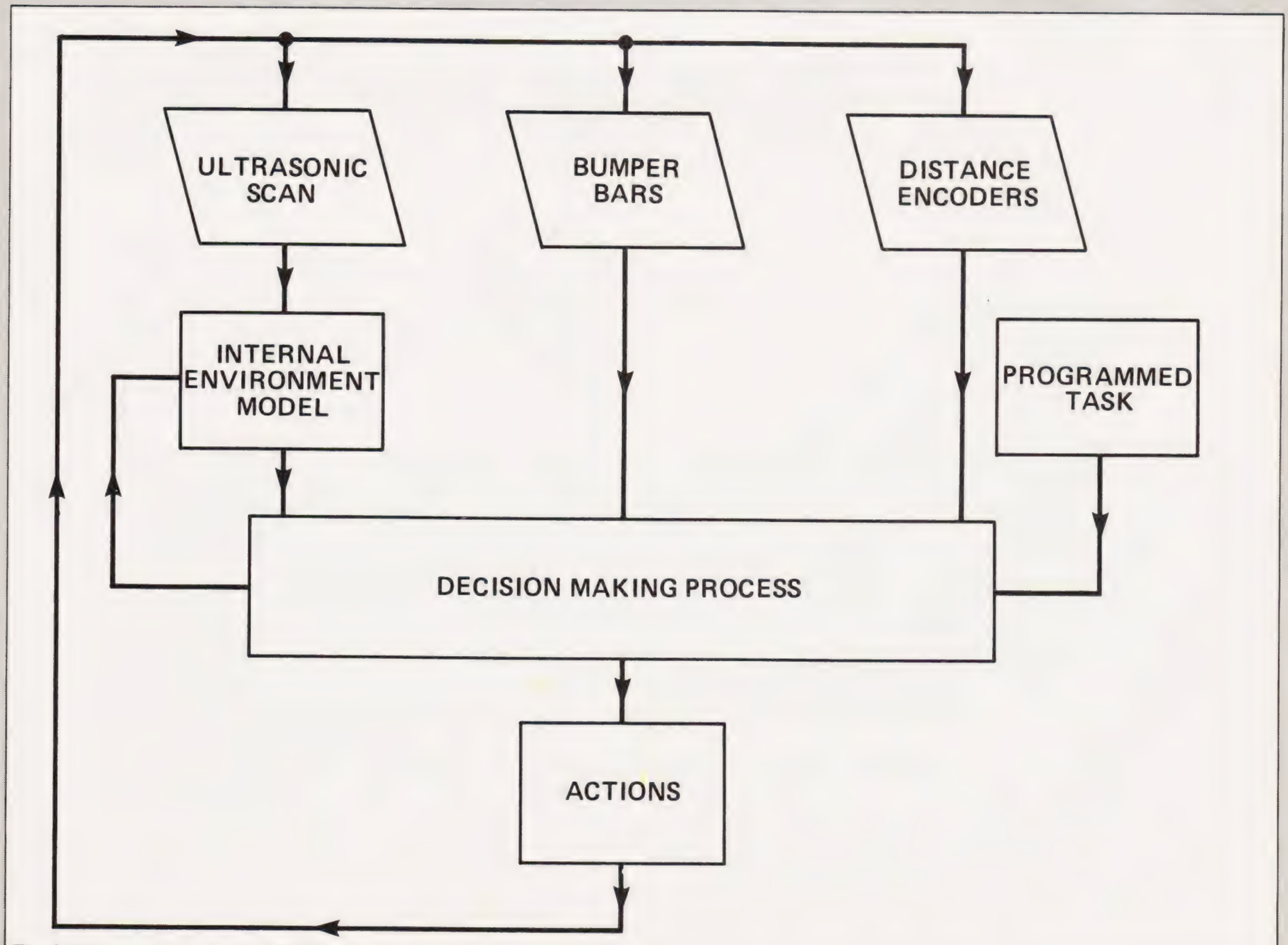


Fig. 2 Schematic for an 'intelligent' robot.

The Hero 1 robot from American manufacturers Heath (picture courtesy Thames TV). On page 32 you can see Prism's latest Topo and Fred personal robots.

ing which way to go! A more practical solution would be to use multiple processors operating simultaneously, each responsible for a separate facet of the robot's senses. At present, parallel processing such as this requires complex, and hence expensive, software and hardware. The decision-making process alone would require a large amount of memory and sophisticated software techniques. It is significant that our poor little robot with only three sources of information to cope with and a simple model to construct is already causing us involved design problems and all it can do is move around in a simple environment and guess where it is!

INTOLERABLE

Of course, all this manipulation of sensory data assumes no errors. To ensure that our robot ends up where it intends, the mechanical parts that control movement must be constructed to fine tolerances. Even high standards of engineering are not enough to smooth out some of the lumps and bumps of the real world that we may literally take in our stride every day. For instance, caster-type wheels have an infuriating habit of pushing you the wrong way. Anyone who has pushed a less than well-oiled supermarket trolly round a crowded store will have experienced this problem. If distances are measured through the revolutions made by the wheel then a skidding or sticking caster is going to throw up a lot of inaccurate data. Orientation is also a major problem for robots working within confined spaces. An error of 0.5 degrees made during a turn means that after travelling 100 metres the robot will be almost 1 metre off course. Practical errors such as these have to be overcome using sophisticated feedback control loops. If the robot were travelling down a straight corridor it would continuously keep a check on its progress by measuring the distance to the adjacent corridor wall. If this measured distance continually increased or decreased then the robot could surmise that its course was not parallel to the wall and make the necessary adjustments. Even the

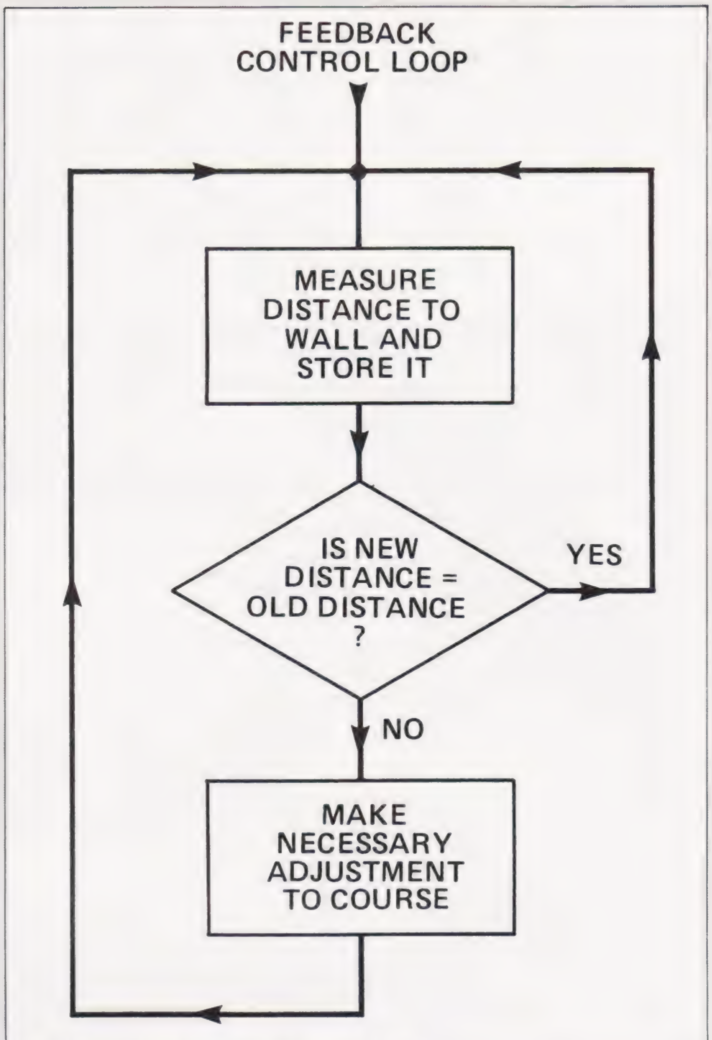


Fig. 3 Flowchart for 'feeling along a wall'.

ultrasonic detectors, however, are not free from error. Flat surfaces can cause the rebounding signals to ricochet at strange angles. The only way that the robot can keep a check on these errors and avoid being misled by them is to draw its information from several sources and cross-check between them.

CONCLUSIONS

In this article we have pointed out a few of the more significant aspects of independent robot movement. To illustrate the problems we have taken the simplest example of a wheeled robot in a flat, single level environment. This type of machine is now a practical proposition and, indeed, a robot similar to that described is currently under construction in the USA by a team of four systems engineers at a company called TRW. Although much money, time and effort is being expended on the project, the robot will not be able to cope with human sized objects in motion and will have no function other than to move about avoiding obstructions. If we now think about the problems thrown up by a multi-level environment involving the negotiation of stairs or an uneven surface, which would mean evolving our robot into a multi-legged creature, the hardware design and construction difficulties pale into insignificance when compared with the task of creating software sophisticated and compact enough to control it! Robots capable of independent movement to carry out tasks in any environment are unlikely in the forseeable future. Such beings would probably require an artificial neural net as complex as our own to cope with the unpredictable nature of our fast-moving, complex world. It is more likely that true free-ranging robots will be first employed in uncluttered environments unnatural to man as in space or under the sea where navigation is more straightforward and the behaviour of obstacles more predictable.

THE ATMOS FOR APPLICATIONS

Samuel Watts

With the microdisc and four colour printer now available, how does the Atmos rate as a complete system?

The Atmos, without any additional peripherals, gives the impression of being a computer in the same group of machines as the Spectrum and the Dragon. That is, a computer intended for home use and, dare I say it, mainly for playing games with. The Spectrum has recently gained more of an air of respectability with the arrival of the long-awaited Microdrives. The Dragon, now called the GEC Dragon to reflect its change in ownership, has followed suit with the availability of its 5¼" disc drives and the powerful OS9 operating system. With this clear trend it seems a natural step for Oric to introduce new peripherals to make the Atmos more attractive for 'serious' applications.

Oric's choice of disc system for the Atmos (and for their earlier computer, the Oric-1) is interesting because it uses a 3" drive from Hitachi. This makes the Atmos plus disc a physically smaller system than would have been necessary if a 5¼" drive had been used but still provides the speed, storage capacity and reliability of a 'real' disc system. In addition to the microdisc, Oric have also taken the unusual step of making the MCP-40 four-colour printer/plotter available in the same colours and styling as the Atmos — so making it their standard printer.

The complete Atmos system; 48K Atmos, microdisc and colour printer/plotter all in black and red cases certainly looks good but how much is it capable of? The best way to answer this question is to describe the features offered by the new peripherals and then try to pinpoint the successes and failures of the overall system.

The first question to clear up is what the reasons are for wanting to replace tape storage by disc storage. It is obvious that both tape and disc can be used to store programs and data as named files. Saving and loading programs from disc differs only in the speed and reliability of operation. The increase in speed is no small advantage. For example, because program saving is so fast, you can afford to save more often during program development and

with the same true for loading, large programs can be written as a number of smaller programs in such a way that only the section actually being used need be loaded into memory. Although speed and reliability are the most obvious differences between tape and disc, there is a more important and fundamental distinction between them when it comes to data handling.

Data that is stored on tape can only be retrieved in the same order that it was written. That is, tape is a 'serial' or 'sequential' storage device. By contrast, data can be retrieved from disc in any order irrespective of how it was written. In other words, disc is a 'random access' storage device. This is due to the different physical forms of tape and disc. Obviously, as data is recorded on tape item by item you have to wind the tape on to reach an item recorded later on the tape. In the case of a disc, the data is recorded in concentric rings and the reading head, the disc equivalent of the tape head, can move to any of the rings of data. This is exactly like the difference between a record player and cassette recorder. It is easy to select a particular track you wish to hear on a record whereas a cassette tape has to be laboriously wound to the correct position. This is also a disinction between true disc drives, 8", 5¼" and 3" and the so called 'stringy floppies' typified by the Sinclair Microdrives. A stringy floppy is essentially a continuous loop of tape that runs past a fixed read/write head. This is obviously a serial access device although software can be used to give the impression of (slow!) random access.

In principle, the Atmos microdisc can be used as a random access device for the storage of data but in practice (see later) this is not possible from Atmos BASIC. This is a great pity as it rules out many sophisticated data processing applications unless you are prepared to use assembler. In fact, the only way that the random access nature of the microdisc shows itself to the BASIC programmer is in the way that many files can be stored on one microdisc and each one loaded just as fast as any other. (Compare this to the situation of storing many files on a cassette tape!)

THE MICRODISC PACKAGE

The Atmos microdisc unit actually consists of five different components:

- The 3" Hitachi disc drive
- The disc controller
- 8K of additional ROM software
- Approximately 11K of RAM software (supplied on microdisc)
- An external power supply

The 3" disc drive is perfectly standard and as far as I can tell, unmodified. It is mounted inside an 11" by 4.5" by 2.75" black and red plastic case (see photo) which also contains the disc controller. The disc controller is constructed on a single PCB and is based on a 1793 chip, interfaced to the Atmos via the expansion port at the rear of the machine. The PCB also contains an 8K ROM (on the review system this was in fact an EPROM) but more of this later. The single drive can be extended by the addition of other 3" drives which, of course, don't need to have a disc controller board mounted inside. A reset switch can be found at the back of the master drive; pressing this causes the system to do a 'cold start'.

The extra 8K or ROM and 11K of RAM-based software contains all the necessary extra commands and extensions to BASIC to allow you to make use of the hardware. The actual method used to add this software to the already rather full Atmos is not explained in the manual but it does state that it resides in the same address area used by the Atmos BASIC ROM. The 8K of ROM software is responsible for getting the system started by loading the 11K of RAM-based software from the 'system' disc into the Atmos. It might surprise you to learn that this RAM-based software doesn't take up any of the standard 48K of the Atmos!

The reason for this is that the 48K Atmos actually has 64K of RAM but normally 16K of it is unusable as it occupies the same set of addresses as the BASIC ROM. When you connect the Atmos microdisc, electronics inside the unit makes this extra 16K of RAM available for use by switching the BASIC ROM in and out as necessary. This sharing of the same set of addresses by selecting which of a number of memory devices is active is usually called 'paging' and it is becoming a common feature of 'upgraded' or 'up market' micros. For example, it is standard on the BBC Micro and now even the Spectrum makes use of it via Interface 1. If you take paging into account you could say that the Atmos plus microdisc produces a machine with approximately 86K of memory!

DOS DISCUSSED

Any software that looks after the functioning of disc drives is usually referred to as a DOS (Disc Operating System) and so the additional ROM- and RAM-based software is collectively called "Oric DOS". The extra commands introduced by Oric DOS are all distinguished from Atmos BASIC commands as they are preceded by an exclamation mark. For example, '!DIR' will produce a listing on the screen of all the files currently on a disc.

Whenever the Atmos encounters an exclamation mark at the start of a command it pages in the DOS to see if it is a request for a disc operation. The main job of any DOS is to organise the storage on the disc into a free area and a collection of named files. All DOS filenames are of the form:

Drive no-name.extension

where 'Drive no' specifies which microdisc drive the file is stored on, 'name' is a string containing up to six letters and 'extension' is a string containing up to three letters. For example, "0-MYFILE.DAT" is a valid file name with 'Drive no' set to 0 and 'name' and 'extension' set to "MYFILE" and "DAT" respectively. When using only one microdisc, there is usually no need to specify on which Drive number a file is stored, DOS will assume that all your files are stored on drive 0.

You might be wondering what the reason is for having the 'extension' part of a file name. The answer is that it is sometimes useful to have a number of files under the same name but with different extensions. For example, PROG1.BAS, PROG1.LST, and PROG1.BIN might all refer to the same program, 'PROG1' in different forms. PROG1.BAS could be a standard BASIC version, PROG1.LST might be a formatted version ready for printing and PROG1.BIN might be a version in machine code. In this sense the 'extension' tells you the type of file and the 'name' identifies the file exactly.

There are a number of occasions where it is useful to be able to specify a number of files with very similar file names. Oric DOS allows you to do this by the use of 'ambiguous' file names. (This idea will be familiar to anyone who has used CP/M or almost any other operating system!) Although this sounds complicated, all you have to remember is that as part of an ambiguous file name '?' stands for any single character and '*' stands for any number of characters. For example, "FILE?.DAT" is an ambiguous file name that refers to all files with names of the form "FILE" followed by any letter or digit, "FILE1", "FILE2" and so on. The name "FILE.*" specifies all files with the name "FILE" irrespective of their extension, "*.DAT" specifies all files with the extension ".DAT" and "*.*" specifies all the files on a disc!

The power and ease of use of

any DOS depends on the range of commands that it offers. The current Oric DOS commands are:

!BACKUP drive no 1 TO drive no 2 This will copy all of the data stored on the disc in 'drive no 1' to the disc in 'drive no 2'. If you only have one microdisc then !BACKUP 0 TO 0 will allow you to copy discs by asking you to repeatedly swap the microdisc in the drive. This is usually referred to as a 'single disc copy'.

!COPY filename 1 TO filename 2, optionlist This command can be used both to copy and to merge files. By specifying different options you can perform a single disc copy, merge files, set the protect status (see later) of the copy and overwrite existing files.

!DEL ambiguous filename This command will delete all of the files that are specified by the 'ambiguous filename'. For example, !DEL "*.DAT" will delete all the files with the extension ".DAT". This command has to be used with care — it can easily wipe out a great deal of work!

!DIR ambiguous filename The !DIR command gives details of all files specified by the 'ambiguous filename'.

!DRIVE drive no This sets the default drive number that will be used if you omit the 'drive no' part of a file name. Unless otherwise specified, the default drive is taken to be 0.

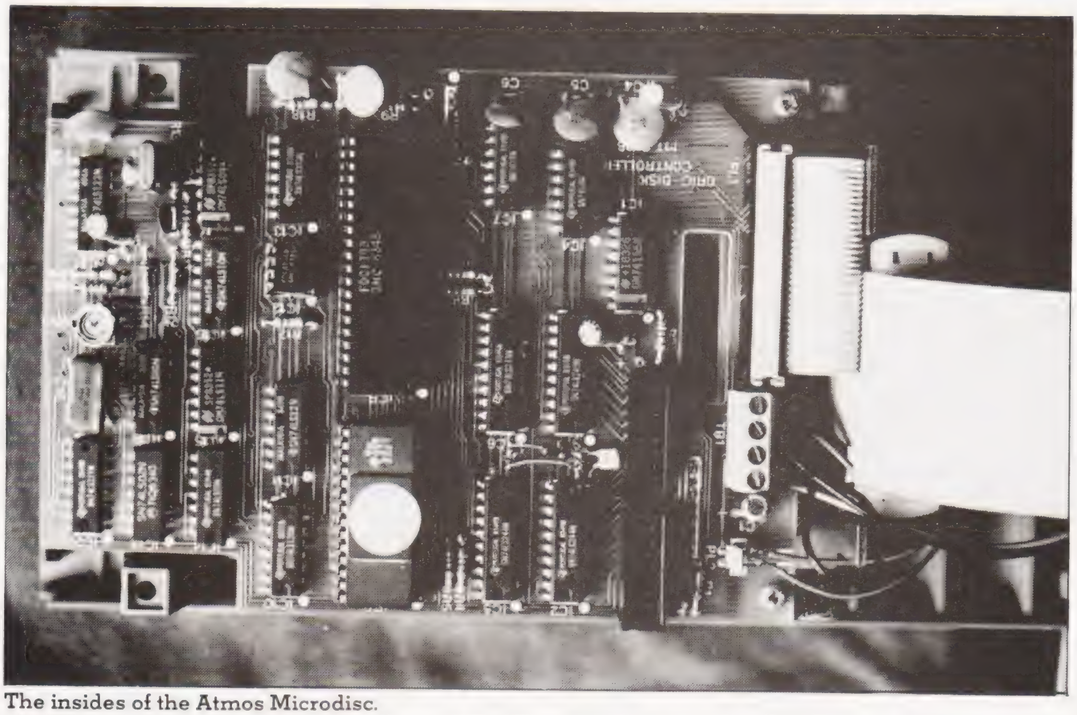
!FORMAT drive no Before a new microdisc can be used it has to be 'formatted' by using the !FORMAT command.

!LOAD filename, option list The !LOAD command is the disc equivalent of CLOAD. It can be used to load a BASIC program or machine code from disc and will automatically start the program running if desired. It can also be used to merge two BASIC programs.

!PROT ambiguous filename, option This command can be used to set or change the status of an existing file or set of files. Using this command a file can be write protected so that it cannot be changed by accident. You can also make a file catalogue protected, that is it will not appear in any directory listing and a user therefore doesn't even know of its existence.

!REN filename1 TO filename2 This command simply changes the name of an existing file called 'filename1' to a new name 'filename2'.

!SAVE filename, options This command is the disc equivalent of CSAVE and can be used to save BASIC programs or areas of memory. An optional auto start address can be specified.



The insides of the Atmos Microdisc.

This set of DOS commands is quite adequate and there are no obvious omissions. In practice the DOS is easy to use and the commands easy to remember. There are still six DOS commands — !OPEN, !CLOSE, !PUT, !GET, !STORE and !RECALL to be described but as these are concerned with manipulating data from Atmos BASIC they are dealt with later.

DISC RESIDENT COMMANDS AND AUTO-BOOT

All of the DOS commands that have been introduced in the previous section are 'memory resident'. That is, the machine code that implements them is loaded into RAM from the system disc as part of the DOS when the Atmos is first switched on. However, the less frequently used DOS commands are 'disc resident' and are stored on disc in named files until they are actually needed. For example, there is an extensive !HELP command that will provide information on how to use any of the DOS commands. This requires a great deal of storage for the explanatory test and so !HELP is implemented as a disc resident command. Typing !HELP causes the file "HELP.COM" to be loaded from disc and run. The general principle is that typing the command "!name" will cause the DOS to search the disc for a file called "name.COM". If it exists, it is loaded and run (if auto run was specified when it was saved).

Obviously the number of disc resident commands can be increased at any time by simply adding new files to the system disc. However, the standard disc resident commands include:

- **!HELP** — gives information on DOS commands
- **!OLD** — restores a program in memory that has been removed using NEW. This command only works if no lines of BASIC have been typed in since the NEW was obeyed.
- **!SYS** — allows the parameters of the disc drive to be changed.

Normally extending a DOS involves using machine code but because a .COM file can be machine code or BASIC, Oric DOS can easily be extended using BASIC. You can write a BASIC program to carry out some task and then save it using the autostart option under a file name using the three letter extension ".COM". Then simply typing "!" followed by the name will cause the program to be loaded and run just like any other disc resident command. This ability to extend the operating system is a powerful tool in the creation of simple-to-use applications systems and it is something that other operating systems would do well to copy.

Another related feature of Oric DOS that is extremely useful is the ability to run a program automatically when the machine is first switched on. This is achieved by the use of a special file called BOOTUP.COM. If this file is present on the system disc it is loaded and if an auto start address has been specified it is run. As in the case of disc resident commands, the BOOTUP.COM file can contain either machine code or BASIC. In practice the BOOTUP.COM file is usually a complete application or games program or contains a menu that offers the user the choice of loading one of a number of programs.

DATA FILES

The commands !STORE and !RECALL are the exact disc equivalents of STORE and RECALL as used with cassette tape. That is,

!STORE array name, filename

will store the array 'array name' on disc in a file called 'filename' and

!RECALL array name, filename

will load the array stored in 'filename' into the array called 'array name'.

Although STORE and RECALL are often useful, there are four completely new commands for handling data files. The !OPEN command is concerned with getting data files ready for processing and comes in two different forms:

!OPEN filename,W

will create a brand new file called 'filename' and get it ready to store data whereas

!OPEN filename,R

will get an existing file ready to be read. In other words, you can OPEN a file for Writing or for Reading.

Once a file is OPEN for writing, data can be stored in it using the !PUT command. The !PUT command can be used to write the contents of numeric variables or string variables to a file. However, you can only !PUT numeric values in the range 0 to 255. You might think that this would severely limit the use of disc data files but this is not so as it is quite easy to convert a numeric variable to a string using STR$ before using !PUT to store it in a file. That is, instead of !PUT A you have to use !PUT STR$(A). You can !PUT more than one variable at a time, for example,

!PUT A$,B,C

but you must read data back in the same order in which it was written out.

The command !GET will read data back from a file OPEN for reading. It follows the same format and suffers the same restrictions as the !PUT command. In particular, if you try to !GET a numeric variable it will return a value only in the range 0 to 255. In general, unless you have a specific reason for using numeric variables in this range it is always better to use !PUT to store strings of digits and use !GET A$:A=VAL(A$) to read and convert them to a full range of numeric values.

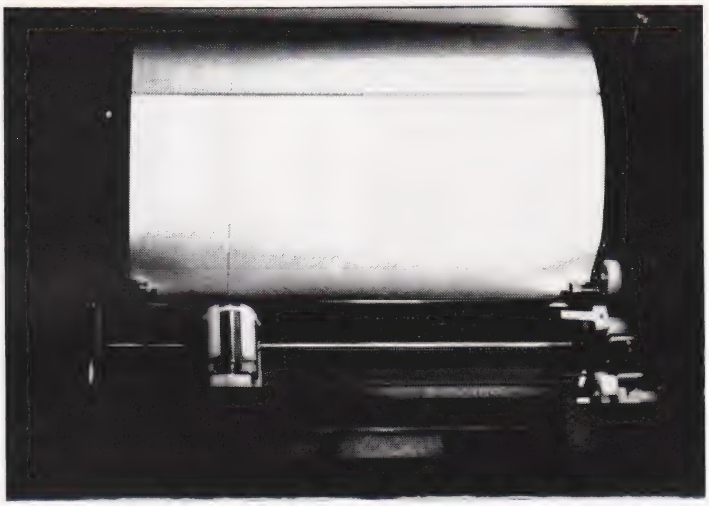
!CLOSE option is a command informing the DOS that you have finished with the files that were OPENed. if 'option' is W then the file that has been OPEN for writing is stored on disc. If 'option' is R then the file that has been OPEN for reading is finished with and another file can be OPENed if required.

ORIC AND ATMOS COMPATABILITY

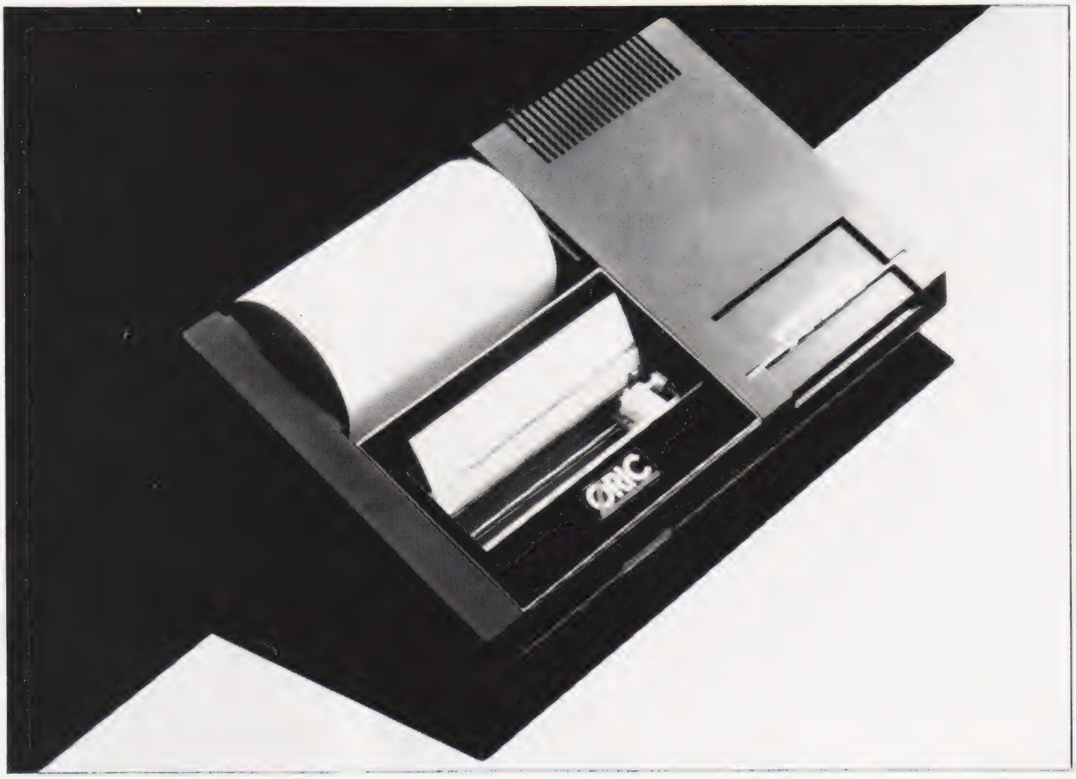
In last month's review of the Atmos I stated that the Oric and the Atmos were completely compatible. As soon as it was too late to alter the script, I discovered three incompatibilities. The first two are minor and concern the way that the TAB and comma work.

On the Oric, TAB(x) had (or rather should have had) the same effect as SPC(x) and the comma always puts three spaces between PRINT items. On the Atmos, TAB(x) takes the print cursor to column x and the comma moves it to the next print zone. Both differences can cause output produced by programs in Oric BASIC to look odd when run on the Atmos.

The third incompatibility is much more serious. To make the text screen co-ordinate system easier to understand, Atmos BASIC starts numbering the columns from the far left starting from zero. This sounds reasonable and identical to the Oric's system until you notice that Oric BASIC ignored the 'protected column' that stored the background attribute code! The result is that Atmos horizontal co-ordinates run from 0 to 39 but on the Oric they run from 0 to 38. The solution is to add one to all the low resolution x co-ordinates in any Oric BASIC program that you want to run on the Atmos.



Above: Close-up of the MCP-40 pen mechanism.
Below: The stylish casing of the printer/plotter.



The commands !OPEN, !PUT, !GET and !CLOSE are remarkably simple when compared to the range of commands that other DOSs make available to their version of BASIC. The thing that worried me when I first started using Oric DOS was that 'simple' might turn out to mean 'restricted'. With Oric DOS, for example, you can only have two files open at any one time, one for reading and one for writing. Most DOSs allow five or six files to be open at the same time! In practice however, Oric DOS proved to me that it is very rare that you NEED to have more than two files open! In the same way, the limitations on !PUT and !GET proved less trouble than I expected. In short, data handling in Oric DOS is limited but it appears able to deal with most of the common situations.

THE MCP-40

The printer chosen by Oric Products to accompany the Atmos and the Oric-1 is the MCP-40 printer/plotter. This device will actually work with any computer with a centronics interface but its high resolution four-colour graphics capability make it a natural companion for the Atmos. Most computer hard copy devices are either printers or plotters but the MCP-40 is both! This dual function is achieved by using any one of four differently coloured ball-

```
 10 HIRES
 15 LPRINT CHR$(18)
 16 DEF FNX(X)=2*X
 17 DEF FNY(Y)=-2*Y
 20 FOR I=1 TO 199 STEP 4
 30 CURSET 0,I,1
 35 LPRINT "M0,";STR$(FNY(I))
 40 DRAW 239-I,-I,1
 45 LPRINT "J";STR$(FNX(239-I));",";STR$
(FNY(-I))
 50 CURSET I,0,1
 55 LPRINT "M";STR$(FNX(I));",0"
 60 DRAW 239-I,I,1
 65 LPRINT "J";STR$(FNX(239-I));",";STR$
(FNY(I))
 70 CURSET 0,199-I,1
 75 LPRINT "M0,";STR$(FNY(199-I))
 80 DRAW 239-1,I,1
 85 LPRINT "J";STR$(FNX(239-I));",";STR$
(FNY(I))
 90 CURSET I,199,1
 95 LPRINT "M";STR$(FNX(I));",";STR$(FNY
(199))
 100 DRAW 239-I,-I,1
 105 LPRINT "J";STR$(FNX(239-I));",";STR
$(FNY(-I))
 110 NEXT I
```

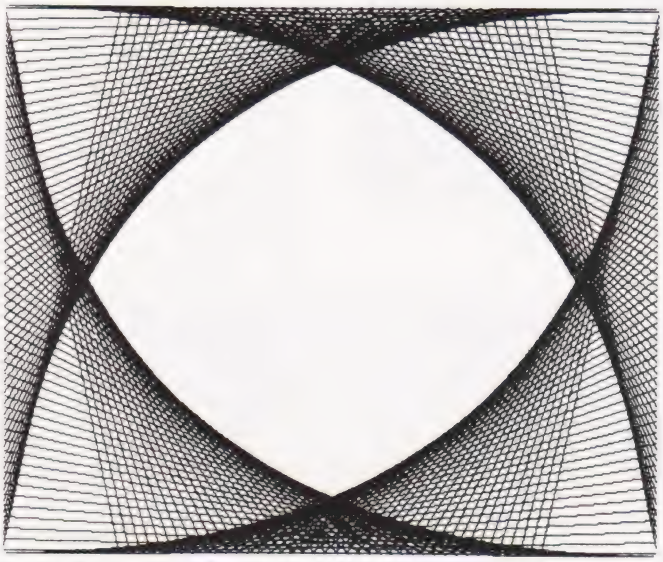


Fig. 1 Samples of the output of the MCP-40 (at about 90% reduction).

TABLE 1: GRAPHICS COMMANDS

Command	Action
Lp	Set line type — p=0 gives solid line p= to 15 give different degrees of dashed/ dotted line
A	Reset printer (to text mode)
H	Home pen to 0,0
I	Initialise — set origin to current pen position
Dx,y	Draw a line from current pen position to the point x,y
Jdx,dy	Draw a line from the current pen position (at x,y) to the point x+dx,y+dy
Mx,y	Move pen to x,y without drawing a line
Rdx,dy	Move the pen from its current position at x,y to x+dx,y+dy
Cn	Change pen n (n=0 to 3)
Sn	Scale the size of text characters n=0 to 63
Qn	Rotote text characters (n=0 to 3)
Ptext	Print 'text' while in graphics mode
Xp,q,r	Draw co-ordinate axis p=0 draws Y axis p=1 draws X axis, q=distance between marks r=number of marks

point pens to draw lines on standard paper. Both the drawing pen and the paper can be moved independently, the pen horizontally and the paper vertically, so making it possible to draw almost any shape.

When the MCP-40 is first switched on it is in text mode and behaves very much like a standard printer. The Atmos BASIC commands, LLIST and LPRINT, perform the same actions as LIST and PRINT except the output is directed to the printer. Apart from these two new commands, the only other complication is the use of some ASCII codes as 'control codes' to do things such as changing the pen colour but even this is a function of 'standard' printers.

In graphics mode, however, any characters that are sent to it are interpreted as graphics commands (see Table 1). The main difference between the graphics commands used by the MCP-40 and the Atmos is that the MCP-40 is essentially a line drawing device whereas the Atmos is a point drawing device. That is, the MCP-40 produces graphics by moving a ball-point pen over paper, producing lines; the Atmos produces lines by drawing a sequence of dots. It is true that you can produce dots on the MCP-40 by drawing very short lines but this is not the most efficient way of using the printer and there is no specific command for plotting a single point. This means that it is very difficult to produce a screen dump program.

The only other important point to bear in mind is that while all of the graphics commands will accept co-ordinates in the range −999 to +999, the actual dimensions of the plotting area are restricted. In the x direction, the restriction is obvious in that the paper is only 480 units wide! Theoretically, the y direction is unlimited — this is generally true for negative values of y because it moves the pen down the paper roll. However, positive values of y can very easily move the pen off the top of the paper and so ruin the output. The solution is of course to always ensure that there is enough paper already fed through the printer to avoid going off the top!

The MCP-40 is great fun to use and its commands make it very easy to produce graphs and charts. However, most programs are written to display their results on the screen using the Atmos high resolution graphics and not being able to 'dump' an existing screen full of graphics to the printer is something of a problem. If you want to write programs that use both the screen and the printer then you will have to bear this in mind as the program is being written and not try to add it as an afterthought. As far as standard listings go, the MCP-40 is very slow at 12 characters per second and although its quality is excellent the narrow paper is a problem if you want to use it as part of a text processor. There is no doubt that the MCP-40 can produce very impressive high resolution graphics but on the whole I would rather settle for a standard dot matrix printer and a screen dump program — unless I had a specific application and a specific program in mind to use with it.

THE COMPLETE WORKS

You might have guessed that as a complete system I am more impressed by the part that the microdisc plays than the MCP-40. The MCP-40 is good, it's just that for many applications a standard dot matrix or even daisy wheel printer might be better. The microdisc makes the Atmos a pleasure to use and apart from not being able to handle random access files from BASIC, I have no real criticisms of it. The use of paging to store the DOS means that the Atmos plus discs has as much user-memory as an Atmos alone and that's not something that many (any?) other micros can claim. Overall, I would say that the investment in the microdisc is money well spent for any Oric or Atmos owner and their availability makes either machine a more attractive proposition.

EXTENDING THE 64'S BASIC PART 3

Tony Cross

Now that we've got the framework ready, it's time to start hanging some new keywords onto it. Here we use some of the integer and floating point maths routines from the BASIC ROM.

Last month I described the 'Extendable BASIC System' which allows you to add new keywords to the existing Commodore 64 BASIC. However, until you've actually written some new keyword routines you can't do a lot with it! So this month, I'm going to show you how to go about writing these routines, and I've included some useful new keywords which you can use in your own version of BASIC.

You may be surprised to learn that writing keyword routines is actually a fairly simple process. This is because most of the really difficult operations, like evaluating expressions and testing for special characters, can already be done by routines within the BASIC ROM. All we need to do is to call the appropriate routines to carry out the operations we require. To be able to do this you need to know a lot more about the contents of the BASIC ROM, and this is one of the things that I will be looking at later. In addition, you need to know how BASIC programs are interpreted — in particular how parameters are extracted from the program lines — and this is where I'm going to start this month.

A LINE AT A TIME

As I explained in an earlier part of this series, a BASIC program consists of one or more separate lines of code. Each line is a self-contained unit — that is, each line contains all the information needed to completely interpret that particular keyword. Consequently a BASIC program is interpreted a line at a time, starting at the beginning of the program and working down. This means that regardless of the length of the program, the Interpreter is only ever looking at one line of code at a time. In addition, every line of code can be completely interpreted by starting at the beginning and working steadily along the line. There is never any need to go back to look at the part of the line again.

So interpreting BASIC is a sequential operation: you start at the beginning and work through character by character until you reach the end of the program. It's rather like reading a book — you start at the first page and 'interpret' each word in a sequential operation until you reach the end of the book.

When I'm reading a book I often use a finger as a pointer to the word I'm currently reading — this stops me from losing my place in the book. The Interpreter does exactly the same thing, except that its pointer is a special two-byte variable called TXTPTR (TeXT PoinTeR).

TXTPTR is located at address $007A/$007B (low byte first as usual). It is a two-byte pointer so that it can point to any memory location, although it is usually restricted to the BASIC program text are ($0800 to $9FFF).

IN AT THE DEEP END

Let's jump straight in at the deep end and see how TXTPTR is used when interpreting a typical line of BASIC. Have a look at the following program line:

```
10 POKE 49152+6*A,INT(B/C)
```

Note: The values of the variables A, B and C are not important. This line of code will be stored in memory as follows:

```
0800  00 18 08 0A 00 97 20 34 39 31 35 32 AA 36 AC 41

0810  2C B5 28 42 AD 43 29 00
```

We've seen before that BASIC programs start at address $0800, and so this is the value that is assigned to TXTPTR when you type RUN.

When the Interpreter reads the null at address $0800 it assumes that it has reached the end of the line. (That's why this byte is always null). The Interpreter 'knows' that the next four bytes are 'special' — ie the first two are a link pointer to the next line, and the next two are the line number bytes. TXTPTR is stepped over these bytes (although their values are stored in case they are needed) leaving it pointing to the first byte of the statement proper.

This first byte (at address $0805) must be either a token or an optional LET (eg 10 A=32 is an optional LET). We've already seen how tokens are dealt with, but if this byte is not a token then the Interpreter assumes that this is an optional LET statement and executes the LET keyword routine. In the example above, however, this byte is the token for POKE and the Interpreter uses it to find the address of the POKE keyword routine. TXTPTR is now stepped on to the first non-space character after the token, ie to the start of the first parameter (at address $0807), and control is then passed to the POKE routine.

The POKE routine 'knows' that following the token byte there should be two parameters separated by a comma. The first of these must be a number (or an expression) which is a decimal integer in the range 0 to 65535, and the second must be a number (or an expression) which is a decimal integer in the range 0 to 255. To evaluate the first of these parameters POKE calls the 'expression evaluaton routine'. This is a very important routine which is fundamental to the working of the whole interpreter, and I'll be looking at it in detail later. Briefly, expression evaluation (or EXPR for short) evaluates an expression (numeric or string) from the current TXTPTR location up to the first delimiter character. Delimiters are characters that mark the end of an expression (eg comma, colon, null, quotes and so on).

So in the example above, EXPR will evaluate everything from address $0807 up to, but not including, the comma at address $0810 (TXTPTR will be left pointing to this comma). I'll be looking at how answers are returned by EXPR later on, for now just accept that an answer is returned and that it can be stored for later use.

The POKE routine now tests the current character to see if it is a comma. It does this by calling a character-checking routine called TSTCOM (TeST for a COMma). TSTCOM checks that the character being pointed to by TXTPTR is a comma, and gives a SYNTAX ERROR message if not. TSTCOM leaves TXTPTR

pointing to the first non-space character after the comma (ie at address $0811).

To evaluate the last parameter POKE again calls EXPR. This time EXPR stops at the null at the end of the line (at address $0817), and returns the evaluated second paramater.

POKE can now load the evaluated memory location with the evaluated value. Having completed its task, POKE returns control to the Interpreter.

The Interpreter now checks that the current character is either a null (end of line) or a colon (end of statement) and gives a SYNTAX ERROR if not. In the example above this character is a null, so all is well and the Interpreter can begin interpreting the next line of code.

EXPRESSION EVALUATION

In the previous example, when the POKE keyword routine wanted to 'read' a parameter from the program line, it called the expression evaluation subroutine with TXTPTR pointing to the start of the parameter. In fact, most parameters are extracted from the program lines in this way. It doesn't matter whether the parameter is a constant, a variable or a complex expression, the expression evaluation subroutine will always return the value of the parameter. Consequently, expression evaluation is easily the most important routine in the Interpreter, so let's look at it in more detail.

The expression evaluation routine itself is an extremely long and complicated routine — far too complex to even begin to explain in the space available here. Using the routine, however, is a fairly simple process because there are a number of different 'entry points' depending on the type of expression you expect to see. Figure 1 is a simplified block diagram of the structure of the expression evaluation subroutine.

There are four main entry points to the routine, three for numeric expressions and one for strings. The first two numeric entry points will only return integer values, so if the expression evaluates to a real number it will be converted to an integer before being returned. In addition, the first entry point will give an ILLEGAL QUANTITY error if the expression evaluates to a number outside the range 0-255 decimal. And the second entry point will give an ILLEGAL QUANTITY error if the expression evaluates to a number outside the range 0-32767 decimal.

The third entry point will return a floating point number in the range −1.7 E 38 to +1.7 E 38. If the expression evaluates to a number outside the range then an OVERFLOW error will be given. (This is actually the maximum numeric range of the machine).

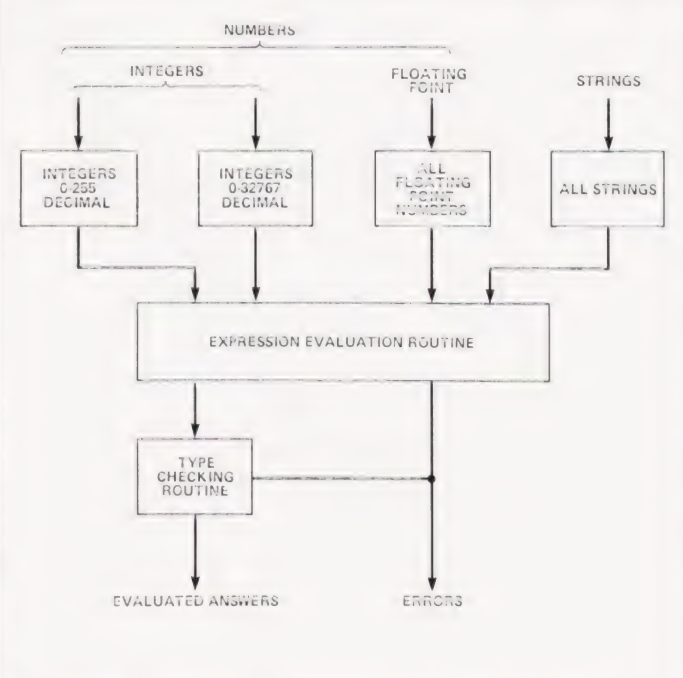


Fig. 1 Block diagram of the expression evaluation routine.

The fourth entry point is used for evaluating all types of string expression (eg A$=B$+C$ or if A$<B$ THEN . . .). If the string expression evaluates to a string which is longer than 255 characters then a STRING TOO LONG error will be given. (I'll be looking at string expression evaluation in a later article).

In all cases, if the expression is invalid in some way (eg A=32+*3 or A$=B$*C$) then a SYNTAX error will be given.

On leaving the routine the value returned is checked to ensure it is of the expected type. If not then a TYPE MISMATCH error will be given. For example, if a floating point number was expected (by entering via the third entry point) and a string was found, then a TYPE MISMATCH error will be given.

THE INTEGER ENTRY POINTS

INCBYT at address $B79B is the main entry point for evaluating integers in the range 0-255 decimal (the first entry point). INCBYT increments TXTPTR and then evaluates the expression found. The integer result, which is returned in the X register, is checked for the range 0-255. If outside this range then an ILLEGAL QUANTITY error will be given. All the CPU registers are modified by this routine and TXTPTR is left pointing to the delimiter character at the end of the expression.

GETBYT at address $B79E is a secondary entry point for evaluating integers in the range 0-255 decimal. It works in exactly the same way as INCBYT except that TXTPTR is not incremented first.

INCINCT at address $B1B2 is the main entry point for evaluating integers in the range 0-32767 decimal (the second entry point). INCINT increments TXTPTR and then evaluates the expression found. The integer result, which is returned in locations $0064/$0065 (high byte in $0064, low byte in $0065), is checked for the range 0-32767. If outside this range then an ILLEGAL QUANTITY error will be given. All the CPU registers are modified by this routine and TXTPTR is left pointing to the delimiter character at the end of the expression.

GETINT at address $B1B5 is a secondary entry point for evaluating integers in the range 0-32767 decimal. It works in exactly the same way as INCINT except that TXTPTR is not incremented first. (It is also possible to evaluate integers in the range 0-65535 decimal, but this involves using the floating point entry point and a special integer conversion routine. I will deal with this when I look at floating point numbers later on).

THE CHARACTER CHECKING ROUTINES

In the example I used earlier, the POKE keyword routine called a routine called TSTCOM to check that the current character (pointed to by TXTPTR) was a comma. If this character had been a comma then TSTCOM would have returned with TXTPTR pointing to the first non-space character after the comma, otherwise a SYNTAX error would have been given.

There are four of these character-checking routines, and they all work the same way (they just check for different characters). TSTCOM at address $AEFD checks for a comma',' ($2C). TSTOPB at address $AEFA checks for an open bracket '(' ($28). TSTCLB at address $AEF7 checks for a close bracket ')' ($29). A more general character-checking routine is TSTCHR at address $AEFF. TSTCHR checks that the current character is the same as the character in the A register.

In all cases, if the test is successful the routine will return with TXTPTR pointing to the first non-space character after the checked character. If the test fails then a SYNTAX error will be given. The CPU A and Y registers are modified by these routines but the X register is unaltered.

TEXT SCANNING ROUTINES

This facility wasn't illustrated by the POKE example I used earlier. In fact, it is fairly rare for you to ever need to move TXTPTR directly since the Interpreter routines adjust it automatically. If you *do* need to move it, however, you can use the same routines that the rest of the Interpreter uses.

The main text scanning routine is called CHRGET at address $0073. It increments TXTPTR and then scans forward along the program line looking for the next non-space character (ie space

TABLE 1

X Register value	Error message
01	Too Many Files
02	File Open
03	File Not Open
04	File Not Found
05	Device Not Present
06	Not Input File
07	Not Output File
08	Missing File Name
09	Illegal Device Number
0A	Next Without For
0B	Syntax
0C	Return Without Gosub
0D	Out Of Data
0E	Illegal Quantity
0F	Overflow
10	Out Of Memory
11	Undef'd Statement
12	Bad Subscript
13	Redim'd Array
14	Division By Zero
15	Illegal Direct
16	Type Mismatch
17	String Too Long
18	File Data
19	Formula Too Complex
1A	Can't Continue
1B	Undef'd Function
1C	Verify
1D	Load
1E	Break

characters are ignored and TXTPTR is stepped over them). On return, the carry flag will be set if a non-numeric character was found, the zero flag will be set if at the end of statement (ie null or colon characters found) and the character found will be in the A register. The CPU X and Y registers are unaltered by this routine and TXTPTR will be left pointing to the character found.

CHRGOT at address $0079 is a secondary entry point to the test scanning routine. It works in exactly the same way as CHRGET except that TXTPTR is not incremented first. (The sharp-eyed ones among you will have noticed that the TXTPTR variable, at address $007A/$007B, is actually contained within the CHRGET routine itself).

DEALING WITH ERRORS

The printing of error messages is another facility that wasn't illustrated by the POKE example I used earlier. Most of the Interpreter routines already print the appropriate error messages automatically. However, there will be occasions when you will want to generate an error message at other times.

There are actually two ways of printing error messages; the simple way which uses the existing BASIC error messages, and a more difficult way which allows you to print your own special error messages in a later article, so this month I'll look at the simple way.

The ERROR routine is at address $A437 and it uses the value in the X register as a pointer into the error message table which is reproduced in Table 1. For example, if the ERROR routine is called with the X register containing $0E, then an ILLEGAL QUANTITY error message will be printed.

In addition, if the error occurs during the running of a program (as opposed to a direct command error), then an IN (line number) clause will be printed — just as in normal error messages.

SOME NEW KEYWORDS

To illustrate the use of some of these routines, here are three new keywords which can be added using the Extendable BASIC System described last month. The three keywords, INK, PAPER and BORDER are presented in Listing 1. The action of these keywords is as follows:

INK c Sets the text pen colour to 'c'. The value of 'c' must be in the range of 0-15 or an ILLEGAL QUANTITY error will be given.

PAPER c Sets the text screen background colour to 'c'. The value of 'c' must be in the range 0-15 or an ILLEGAL QUANTITY error will be given.

BORDER c Sets the text screen border colour to 'c'. The value of 'c' must be in the range 0-15 or an ILLEGAL QUANTITY error will be given.

Looking at Listing 1 you can see the three short routines for each of the new keywords. I chose the start address for the first keyword ($C416) simply because it was the next available location following the Extendable BASIC System software. You can actually locate these routines anywhere you like.

The INK routine first checks the value in the X register. (Remember from last month, the Extendable BASIC System loads the X register with $0 if the keyword was used as a statement, and $FF if it was used as a function). Since the INK keyword can only be used as a statement, the X register must contain $0, or a SYNTAX error message will be printed. (I've actually cheated a bit here, because I haven't used the ERROR routine to print the syntax error message. Instead I'm using a routine at address $AF08 which exists just to print the SYNTAX error message).

Assuming that the X register does contain $0, the INK routine next calls GETBYT to extract the colour parameter. (I have called GETBYT because the Extendable BASIC System leaves TXTPTR pointing to the first non-space character after the keyword, ie the start of the colour parameter, and GETBYT does not increment TXTPTR first).

The colour parameter is then checked for range. (Remember, the parameter is now in the X register). GETBYT ensures that the colour value is not negative, so it is only necessary to check that it is less than 16. The 'compare X with $10' instruction takes care of this, leaving the carry flag set if the colour is greater than or equal to 16. If the colour value is out of range, then the X register is loaded with $0E and the ERROR routine is called to print an ILLEGAL QUANTITY error message. (The 'return from subroutine' instruction at the end of the ERROR routine will return control to the Editor).

Assuming that the colour value is in range, all that now remains is to load it into the current character colour variable at address $0286.

The 'return from subroutine' instruction then returns control back to the Interpreter. (Notice that I haven't checked that we

```
033C               !##################################
033C               !#                                #
033C               !#    INK-PAPER-BORDER KEYWORDS   #
033C               !#                                #
033C               !#     VERSION 1.0 -- 14/12/83    #
033C               !#                                #
033C               !# COPYRIGHT (C)  A.L.CROSS 1983  #
033C               !#                                #
033C               !##################################
033C               !
033C               !
033C               !VARIABLES AND EQUATES
033C
033C               GETBYT      =  $B79E
033C               COLVAR      =  $0286
033C               BDRVAR      =  $D020
033C               PAPVAR      =  $D021
033C               ERRORS      =  $A437
033C
C416               *=$C416
C416               !
C416               !
C416 E000   INK          CPX #$00                   !CHECK STATEMENT FLAG
C418 F003                BEQ INKOK
C41A 4C08AF SYNERR       JMP $AF08                  !SYNTAX ERROR
C41D 209EB7 INKOK        JSR GETBYT                 !INK COLOUR
C420 E010                CPX #$10                   !IS IT => 16 ?
C422 9005                BCC INKRNG
C424 A20E   QTYERR       LDX #$0E                   !ILLEGAL QUANTITY
C426 4C37A4              JMP ERRORS
C429 8E8602 INKRNG       STX COLVAR                 !SET INK COLOUR
C42C 60                  RTS
C42D               !
C42D               !
C42D E000   PAPER        CPX #$00                   !CHECK STATEMENT FLAG
C42F D0E9                BNE SYNERR
C431 209EB7              JSR GETBYT                 !PAPER COLOUR
C434 E010                CPX #$10                   !IS IT => 16 ?
C436 B0EC                BCS QTYERR
C438 8E21D0              STX PAPVAR                 !SET PAPER COLOUR
C43B 60                  RTS
C43C               !
C43C               !
C43C E000   BORDER       CPX #$00                   !CHECK STATEMENT FLAG
C43E D0DA                BNE SYNERR
C440 209EB7              JSR GETBYT                 !BORDER COLOUR
C443 E010                CPX #$10                   !IS IT => 16 ?
C445 B0DD                BCS QTYERR
C447 8E20D0              STX BDRVAR                 !SET BORDER COLOUR
C44A 60                  RTS
```

Listing 1. The INK, PAPER, and BORDER commands.

are actually at the end of a statement, nor have I checked that there aren't any extra invalid parameters. This is because the Interpreter does this checking for itself and gives a SYNTAX error if the current character (pointed to by TXTPTR) is not a null or a colon).

The PAPER and BORDER keywords are almost identical to the INK keyword. The only differences are the locations of the current pen colour, the current screen colour and the current border colour variables.

FLOATING POINT NUMBERS

Representing integer numbers in a computer is a fairly simple process, because there is a binary equivalent for every decimal integer. A number like 457 can be easily stored in two bytes as 00000001 11001001 (or 1C9 in hex). However, a real number, like 12.36, presents us with a problem because the integer method of storing numbers cannot cope with the decimal fraction (the .36). The Commodore 64's BASIC (like most other BASICs) gets round this problem by storing real numbers in a completely different way. It uses a representation called 'floating point format'.

Floating point format can, at least in theory, represent any real number, so it is used as the standard method of number representation in the Commodore 64. All the maths routines and the trig functions work on floating point numbers (actually, so do the integers — they are converted to floating point format first, and then converted back to integers after the calculations have been performed! — More about this later). So, to make the most efficient use of the Commodore BASIC maths routines you need to understand how numbers are represented in floating point format.

FLOATING POINT FORMAT

Floating point format is very similar to the more common 'scientific notation' used on many calculators. In floating point format the number is split into two parts called the MANTISSA and the EXPONENT.

The mantissa holds the digits which make up the number, with the decimal point always assumed to be in front of the first digit. For example, the mantissa of the number 12.36 would be .1236. The exponent holds the number of times the decimal point must be moved to 'recover' the real number. The exponent of the number 12.36 would be 2. So, quite simply the real number 12.36 is represented as 0.1236 * 10↑2 — where 0.1236 is the mantissa and 10↑2 is the exponent (decimal exponents are always 10 'to the power something', and so the exponent can be simply represented as 2).

Now consider the number 0.001236. How can this be represented in floating point format? Well, we know that the mantissa must be .1236 as before — but what value is the exponent? As I am sure you have worked out, the exponent must be −2. So, the number 0.001236 is represented as 0.1236 * 10↑(−2).

The sign of the exponent, therefore, indicates the direction in which the decimal point must be moved. A negative exponent means move the decimal point to the left, and a positive exponent means move the decimal point to the right.

Now take a look at the number −12.36. How is this represented in floating point format? Well, we know that the exponent will be 2, so the mantissa must be −.1236. The number −12.36 then, is represented as −0.1236 * 10↑2.

The sign of the mantissa, therefore, indicates the sign of the real number.

You can now see that for each of the two parts of a floating point number (the mantissa and the exponent) we must store two pieces of information: the numerical value and the sign. Let's look at the exponent first. We already know that a negative exponent means move the decimal point left, and a positive exponent means move the decimal point right. In addition, an exponent of 0 is taken to mean that the whole of the number is 0. (Even though any number to the Power 0 is really 1). This makes testing for the zero condition much easier.

TABLE 2

Decimal Value	Floating Point Value: Exponent	Mantissa
0	00	Don't care
1	81	00 00 00 00
2	82	00 00 00 00
3	82	40 00 00 00
−3	82	C0 00 00 00
0.25	7F	00 00 00 00
−0.25	7F	80 00 00 00

The high bit of the exponent is used to indicate the sign of the exponent, and the other seven bits are used to indicate the numerical value of the exponent. If the high bit is set (1) the exponent is positive; if the high bit is reset (0) the exponent is negative.

Now on to the mantissa. The logical way to store the sign of the mantissa is to use the first bit — ie the high bit of the first byte of the mantissa. However, this could create a problem because the high bit of the first byte is also part of the number! Actually we *can* use this bit for the sign with no problems at all because of the way in which the mantissa is stored.

In an integer binary number the 'value' of each bit position is twice the value of the one before it, moving from right to left (away from the decimal point), thus:

Bit value 32 16 8 4 2 1.

So the binary number 101 has a decimal value of 1 + 4 = 5. (Assuming that all unspecified higher valued bits are zero, which is the normal convention).

In the mantissa, however, things are a little different. We must first use the exponent byte to fix the decimal point (ie move it left or right as appropriate). All bits to the left of the decimal point then have the same 'values' as those in integer binary numbers (ie 1,2,4,8 etc). For bits to the right of the decimal point, however (the fractional bits) we count from left to right (but still away from the decimal point). The 'value' of each bit is half the value of the one before it, starting with the value 1/2, thus:

Bit value ·1/2 1/4 1/8 1/16 1/32

So a mantissa number of .10011 has a value of 1/2 + 1/16 + 1/32 = 0.59375. (Assuming that all unspecified lower valued bits are zero, which is the normal convention).

Now, because we are using an exponent byte to indicate how far to move the decimal point, we can arrange the mantissa such that the first bit is always set (1). Knowing this we can now use this bit to indicate the sign of the mantissa. This is done by reading it first as a sign bit, and then setting it to 1 before reading the value of the mantissa. If the sign bit is set (1) the mantissa is negative; if the sign bit is reset (0) the mantissa is positive. (Note: This is the opposite way round to the exponent sign bit!).

For example, assuming that the exponent byte is 81 hex (ie move the decimal point one place to the right) then a mantissa value of .110101 will be a negative number (sign bit set) with a value of 1 + 1/2 + 1/8 + 1/32 = −1.65625: whereas a mantissa value of .010101 will be a positive number (sign bit reset) with a value of 1 + 1/2 + 1/8 + 1/32 = −1.65265. (Note: the sign bit is set to 1 and the decimal point position is fixed, before the mantissa value is calculated).

The accuracy of floating point numbers depends on the length of the mantissa. On the Commodore 64 the mantissa is four bytes long — giving an accuracy of better than one part in four thousand million! (It is actually one part in 2↑32 or one part in 4.294967296 * 10↑9).

The range of numbers that can be represented in floating point format depends on the size of the exponent. With a seven bit exponent and a sign bit the range is 2.938736 * 10↑(−39) to 1.701412 * 10↑38 (or 2↑(−128) to 2↑−127). To help make all this a little clearer, Table 2 contains some typical floating point values.

THE FLOATING POINT ACCUMULATORS

The Commodore 64's BASIC uses two floating point accumulators — called 'floating point accumulator 1' (FPA1) and 'floating point accumulator 2' (FPA2). They are used to hold the values that are currently being operated on, with the result of the calculation always returned in FPA1. In general all calculations are of the form:

FPA1 = FPA2 'operator' FPA1

For example, the calculation 12.34 − 2.7681 would be performed with 12.34 in FPA2 and 2.7681 in FPA1. The answer (9.5719) will be returned in FPA1. Even very complex expressions are evaluated this way — the answer being 'built up' by performing the appropriate calculations one at a time — but always using FPA1 and FPA2.

The floating point accumulators use a slightly different format to that used for storing floating point numbers in memory. FPA1 and FPA2 are each six bytes long, the extra byte is a sign byte which holds the sign of the mantissa. This means that the high bit of the first byte of the mantissa is *always* set (1).

FPA1 is located at addresses $61 and $66. The exponent byte is in $61, the four mantissa bytes are in $62 to $65 ($62 is the high byte) and the sign byte is in $66.

FPA2 is located at addresses $69 to $6E. The exponent byte is in $69, the four mantissa bytes are in $6A to $6D ($6A is the high byte) and the sign byte is in $6E.

In addition, FPA1 has an extra byte called a 'rounding byte' and it is used to increase the accuracy of the calculations. The rounding byte is located at address $70; however its action is largely 'automatic' and it is most unlikely that you will ever need to worry about it.

Figure 2 shows how the floating point accumulators are organised.

EVALUATING REAL NUMBERS

Earlier I showed you how to extract integer parameters from a program line using INCBYT and INCINT. Naturally, there is a similar routine for extracting real numbers — it is called NUMEXP (NUMeric EXPression) and it is located at address $AD8A. NUMEXP evaluates the numeric expression pointed to by TXTPTR (TXTPTR is *not* incremented first). The floating point result, which is returned in FPA1, must be within the numeric range of the machine (see earlier). If it isn't, an OVERFLOW error will be given. NUMEXP leaves TXTPTR pointing to the delimiter character and modifies all of the registers.

Having extracted a floating point number from a BASIC line you will probably want to store it somewhere for use later. The ROM routine which does this is called FPA1XY (FPA1 to the XY location) and it is located at address $BBD4. FPA1XY first uses the rounding byte to 'round' FPA1 (you don't need to worry about this — it only slightly modifies FPA1 to improve its accuracy) and then copies the contents of FPA1 to the five memory bytes pointed to by the X/Y registers (high byte in Y, low byte in X). The sign byte ($66) is NOT copied across but the sign bit of the mantissa (high bit of the first byte remember) is set or reset according to the state of the sign byte. The contents of FPA1 and the X register are not changed by this routine (except for the rounding) but the A and Y registers are modified.

If you only want to move the contents of FPA1 to FPA2 then the MOV1T2 routine (MOVe fpa1 To fpa2) at address $BC0C is the one you need. MOV1T2 'rounds' FPA1 and then copies it across to FPA2 (including the sign byte). FPA1 and the Y register are unchanged (except for the rounding) but the A and X registers are modified.

Naturally there is a routine which does the opposite of MOV1T2 — it is called MOV2T1 (MOVe fpa2 To fpa1) and it is located at address $BBFC. MOV2T1 copies the contents of FPA2 to FPA1 (including the sign byte). The rounding byte of FPA1 ($70) is reset but FPA2 remains unchanged as does the Y register. The A and X registers, however, are modified by this routine.

The last two floating point 'move' routines are for loading the floating point accumulators from memory (recovering values stored by FPA1XY for example). AYFPA1 (move AY contents to FPA1), which is located at address $BBA2, copies the contens of the five bytes pointed to by the A/Y registers (high byte in Y, low byte in A) into FPA1. The first byte pointed to will go into the FPA1 exponent byte, the next will be the byte of the mantissa, and so on. The sign byte of FPA1 ($66) will be set according to the state of the mantissa sign bit, and this bit will then be set to 1. The contents of FPA2 and the five bytes of memory pointed to by the A/Y registers are unchanged (as is the X register) and the A and Y registers will be modified.

AYFPA2 (move AY contents to FPA2), which is located at address $BA8C, is similar to AYFPA1 except that the five bytes pointed to by the A/Y registers are copied into FPA2. The contents of FPA1 and the five bytes of memory pointed to by the A/Y registers are unchanged (as is the X register) but the A and Y registers will be modified.

THE FLOATING POINT MATHS ROUTINES

As I said earlier, all calculations are performed on numbers in FPA1 and FPA2, using the general form FPA1 = FPA2 'operator' FPA1. The floating point maths routines however, also have an additional, more useful, format of:

FPA1 = contents of A/Y location 'operator' FPA1.

Of course, the only difference is that these routines load FPA2 from the contents of A/Y first (using AYFPA2). It saves us from having to do it ourselves though!

ADDAY (ADD fpa1 to the contents of A/Y) at location $B867, is the floating point addition routine. It adds the contents of the five bytes of memory pointed to by the A/Y registers (high byte in Y, low byte in A) to the contents of FPA1. The result is returned in FPA1 leaving the five bytes of memory unchanged. FPA2 and all the registers are modified.

ADD, at location $B86A, is a secondary entry point to the addition routine. It adds the contents of FPA1 to the contents of FPA2. The result is returned in FPA1 leaving FPA2 unchanged. All the registers are modified. (Note: on entry to the ADD routine the zero flag must reflect the exponent of FPA1. This can be simply arranged by using a LDA $61 instruction first).

SUBAY (SUBtract fpa1 from the contents of A/Y) at location $B850, is the floating point subtraction routine. It subtracts FPA1 from the contents of the five bytes of memory pointed to by the A/Y registers (high byte in Y, low byte in A). The result is returned in FPA1, leaving the five bytes of memory unchanged. FPA2 and all the registers are modified.

SUB, at location $B853, is a secondary entry point to the

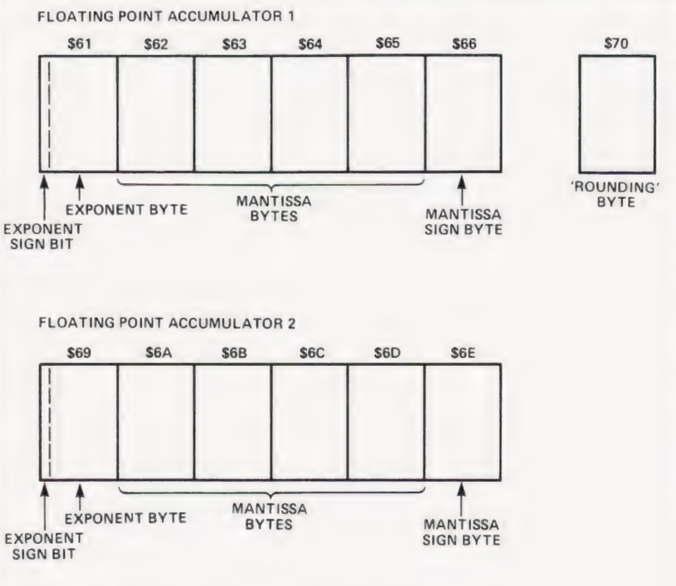


Fig. 2 The memory locations of the floating point accumulators.

subtraction routine. It subtracts the contents of FPA1 leaving FPA2 unchanged. All the registers are modified.

MULTAY (MULTiply the contents of A/Y by fpa1) at location $BA28, is the floating point multiplication routine. It multiplies the contents of the five bytes of memory pointed to by the A/Y registers (high byte in Y, low byte in A) by the contents of FPA1. The result is returned in FPA1, leaving the five bytes of memory unchanged. FPA2 and all the registers are modified.

MULT, at location $BA2B, is a secondary entry point to the floating point multiply routine. It multiplies the contents of FPA2 by the contents of FPA1. The result is returned in FPA1 leaving FPA2 unchanged. All the registers are modified. (Note: on entry to the MULT routine the zero flag must reflect the exponent of FPA1 — just like the ADD routine).

DIVAY (DIVide the contens of A/Y by fpa1) at location $BB0F, is the floating point division routine. It divides the contents of the five bytes of memory pointed to by the A/Y registers (high byte in Y, low byte A) by the contents of FPA1. The result is returned in FPA1, leaving the five bytes of memory unchanged. FPA2 and all the registers are modified.

DIV, at location $BB12, is a secondary entry point to the division routine. It divides the contents of FPA2 by the contents of FPA1. The result is returned in FPA1 leaving FPA2 unchanged. All the registers are modified.
(Note: on entry to the DIV routine the zero flag must reflect the exponent of FPA1 — just like the ADD and MULT routines).

FLOATING POINT TO INTEGER CONVERSIONS

All the floating point maths routines return their values in FPA1. However, there may be occasions when you need to return a binary integer value. If this is the case then there are one or two conversion routines to make this job easier.

CVTINT (ConVerT to INTeger) at location $BC9B, converts a floating point number in FPA1 to a four byte signed integer in locations $62 to $65 (highest byte in $62, lowest byte in $65). FPA1 and all the registers are modified by this routine.

SGNINT (convert to a SiGNed INTeger) at location $B1BF, converts a floating point number in FPA1 to a two byte signed integer (in the range −32768 to 32767) in locations $14/$15 (high byte in $15, low byte in $14). FPA1 and all the registers are modified by this routine.

$POSINT (convert to a POSitive INTeger) at location $B7F7, converts a floating point number in FPA1 to a two byte unsigned integer (in the range 0-65535) in locations $14/$15 (high byte in $15, low byte in $14). FPA1 and all the registers are modified.

If you want to convert a two byte signed integer (in the range −32768 to 3277) to a floating point number, then CVTFPN (ConVerT to Floating Point Number) at location $B391, is the one you want. CVTFPN converts a two byte signed integer in the A/Y registers (high byte in Y, low byte in A) into a floating point number in FPA1.

Although this is the only integer to floating point conversion routine in the ROM, there are some simple techniques for converting larger integers to floating point, and I'll be looking at one of them later.

SOME NEW FLOATING POINT KEYWORDS

The new keywords for this section are DEEK and DOKE. They are similar to the existing PEEK and POKE keywords except that DEEK and DOKE work on numbers in the range 0-65535. This means that they must access two memory locations (DEEK is a double PEEK and DOKE is a double POKE).

DOKE memory location, value DOKE loads the contents of 'memory location' and 'memory location + 1' with the 16-bit value 'value'. Both values must be integers in the range 0-65535 or an ILLEGAL QUANTITY error will be given.

DEEK (memory location) DEEK returns the value contained in 'memory location and 'memory location + 1'. The memory location must be an integer in the range 0-65535 or an ILLEGAL QUANTITY error will be given.

```
033C             !##################################
033C             !#    DEEK AND DOKE  KEYWORDS     #
033C             !#                                #
033C             !#     VERSION 1.0 -- 9/1/84      #
033C             !#                                #
033C             !# COPYRIGHT (C)  A.L.CROSS 1983 #
033C             !##################################
033C             !
033C             !
C44B             *=$C44B
C44B             !
C44B             !
C44B             !VARIABLES AND EQUATES
C44B             !
C44B             CHKSTK       =  $A3FB
C44B             NUMEXP       =  $AD8A
C44B             POSINT       =  $B7F7
C44B             TSTCOM       =  $AEFD
C44B             TSTOPB       =  $AEFA
C44B             TSTCLB       =  $AEF7
C44B             CVTFPN       =  $B391
C44B             MULTAY       =  $BA28
C44B             MOV1T2       =  $BC0C
C44B             ADD          =  $B86A
C44B             !
C44B             !
C44B E000   DOKE        CPX #$0               !CHECK STATEMENT FLAG
C44D F003               BEQ DOKEOK
C44F 4C08AF SYNERR      JMP $AF08             !SYNTAX ERROR
C452 A901   DOKEOK      LDA #$01              !2 SLOTS WANTED
C454 20FBA3             JSR CHKSTK            !ON THE STACK
C457 208AAD             JSR NUMEXP            !GET ADDRESS
C45A 20F7B7             JSR POSINT            !CONVERT TO INT
C45D A514               LDA $14               !LOW ADDRESS BYTE
C45F 48                 PHA                   !SAVE IT
C460 A515               LDA $15               !HIGH ADDRESS BYTE
C462 48                 PHA                   !SAVE IT
C463 20FDAE             JSR TSTCOM            !CHECK FOR COMMA
C466 208AAD             JSR NUMEXP            !GET VALUE
C469 20F7B7             JSR POSINT            !CONVERT TO INT
C46C A615               LDX $15               !HIGH VALUE BYTE
C46E A414               LDY $14               !LOW VALUE BYTE
C470 68                 PLA                   !POP HIGH ADDRESS BYTE
C471 8515               STA $15               !AND PUT IT BACK
C473 68                 PLA                   !POP LOW ADDRESS BYTE
C474 8514               STA $14               !AND PUT IT BACK
C476 98                 TYA                   !LOW VALUE IN A
C477 A000               LDY #$0               !INITIALISE INDEX
C479 9114               STA ($14),Y           !SAVE LOW VALUE BYTE
C47B 8A                 TXA                   !HIGH VALUE IN A
C47C C8                 INY
C47D 9114               STA ($14),Y           !SAVE HIGH VALUE
C47F 60                 RTS
C480             !
C480             !
C480 E0FF   DEEK        CPX #$FF              !CHECK FUNCTION FLAG
C482 D0CB               BNE SYNERR
C484 20FAAE             JSR TSTOPB            !CHECK FIRST BRACKET
C487 208AAD             JSR NUMEXP            !GET ADDRESS
C48A 20F7B7             JSR POSINT            !CONVERT TO INTEGER
C48D 20F7AE             JSR TSTCLB            !CHECK SECOND BRACKET
C490 A001               LDY #$01
C492 B114               LDA ($14),Y           !GET HIGH BYTE
C494 A8                 TAY
C495 A900               LDA #$00
C497 2091B3             JSR CVTFPN            !CONVERT TO FPN
C49A A9B3               LDA #<CONST           !MULTIPLY HIGH
C49C A0C4               LDY #>CONST           !BYTE BY 256
C49E 2028BA             JSR MULTAY
C4A1 200CBC             JSR MOV1T2            !PUT IT IN FPA2
C4A4 A000               LDY #$00
C4A6 B114               LDA ($14),Y           !GET LOW BYTE
C4A8 A8                 TAY
C4A9 A900               LDA #$00
C4AB 2091B3             JSR CVTFPN            !CONVERT TO FPN
C4AE A561               LDA $61               !GET FPA1 EXPONENT
C4B0 4C6AB8             JMP ADD               !ADD FPA2 TO FPA1
C4B3             !
C4B3             !
C4B3             !256 IN FLOATING POINT FORMAT
C4B3             !
C4B3 890000 CONST       BYT $89,$00,$00,$00,$00
```

Listing 2. The DEEK and DOKE keywords.

The full listings for DEEK and DOKE are given in Listing 2. I chose the start address ($C44B) to follow on from the INK, PAPER, BORDER keywords presented earlier, but DEEK and DOKE can be located anywhere.

THE DOKE KEYWORD

The DOKE keyword is a statement because it performs an action, so the first thing to do is to check the statement flag. Assuming that this is OK, we must now check that there is enough space left on the stack (DOKE needs two 'slots'). This can be checked by calling a special routine called CHKSTK (CHecK STacK) at location $A3FB. CHKSTK checks that there are 2 * A bytes left on the stack. An OUT OF MEMORY error will be given if not (CHKSTK modifies the A and X registers but leaves the Y register unchanged).

If all is well with the stack then we can extract the first parameter (the memory location). This must be an integer in the range 0-65535; however, there are no routines in the ROM to extract a value of this type directly. The simple solution is to extract it as a floating point number (using NUMEXP) and then convert it to an integer (using POSINT). The value returned, in locations $14/$15, is the address of the first memory location (where the low 'value' byte must go). We will be using these locations again to extract the value parameter, so the memory location values are saved on the stack.

Now we can use the same technique (NUMEXP and POSINT) to extract the 'value' parameter, and this is returned in locations \$14/\$15 (low byte in \$14).

We now need to temporarily store the high and low bytes of the 'value' in the X and Y registers (high byte in X) and then restore the 'memory location' parameter back into loctions \$14/\$15. (This is so that we can use it as an indirect pointer to the memory location).

By transferring the low value byte to the A register (TXA) and loading Y with 0 we can use the indirect STA (\$14),Y instruction to load the low byte of the value into the first memory location. It is now a simple matter to increment Y, transfer the high value byte to the A register (TYA) and use the same indirect STA instruction to load the high value byte into the next memory location.

The RTS instruction at the end of the routine returns control to the Interpreter.

THE DEEK KEYWORD

The DEEK keyword is a function because it returns a value, so the first thing to do is to check the function flag. This will be \$FF if DEEK has been called 'legally'. Next we must check for an open bracket character (by calling TSTOPB), and then use the NUMEXP/POSINT technique to extract the memory location parameter. The final check on the program line is for the close bracket (by calling TSTCLB).

We can now read the two memory location bytes using the indirect LDA (\$14),Y instruction, with Y containing 0 or 1 depending on which byte is being read. This 16-bit value needs to be converted from an unsigned integer (in the range 0-65535) to a floating point number. (This is because functions *always* return their values in FPA1). As I mentioned earlier, there are no ROM routines to perform this conversion, so it must be done as the values are read. The point to remember is that DOKE stores the 'value' low byte first, so the 16-bit value can be found by:

16 bit value = low byte + 256*high byte

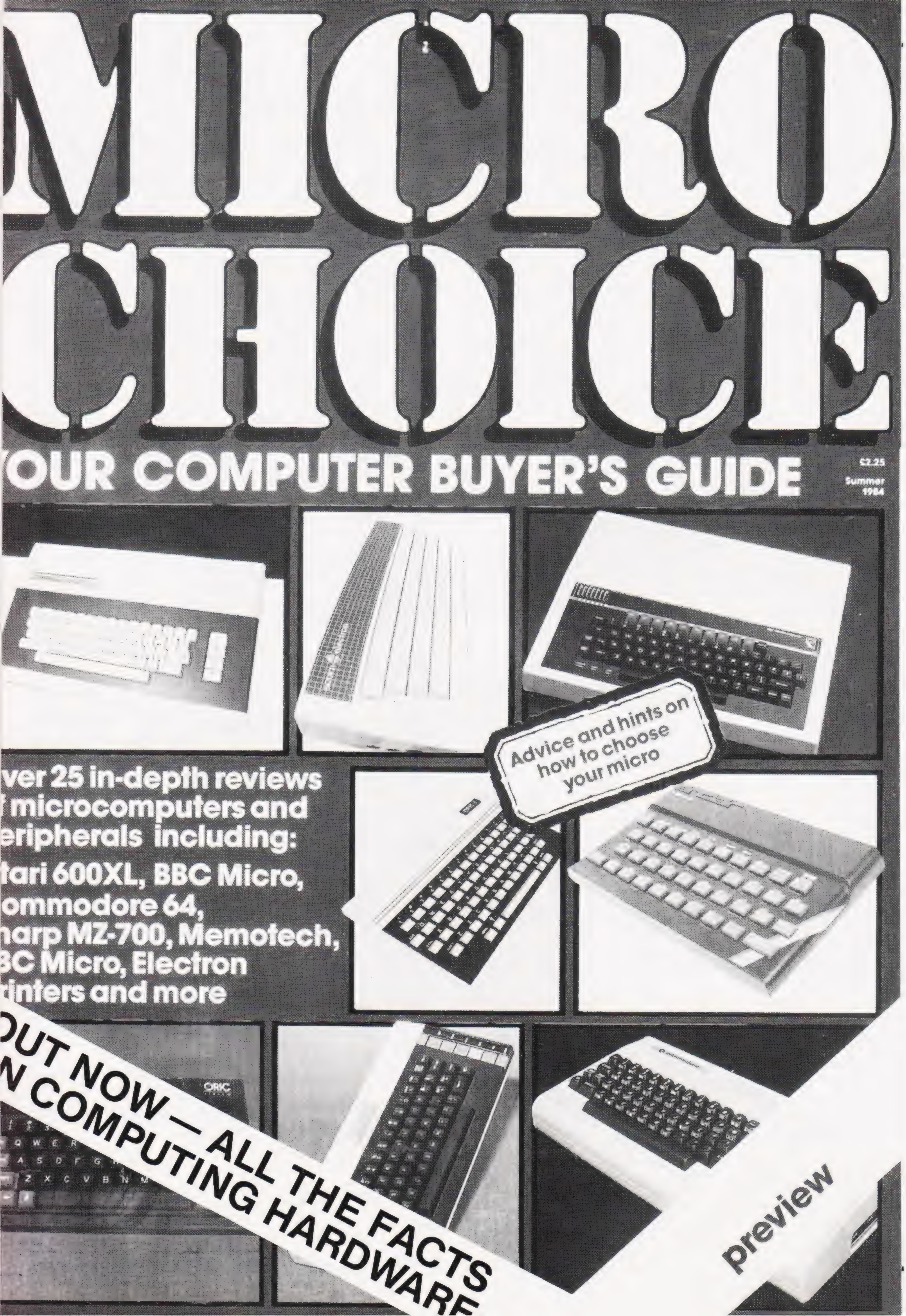
This 16-bit value must be returned in FPA1, so it makes sense to perform all the calculations in floating point format.

The high value byte is read first and then transferred to the Y register. The A register is loaded with 0 and the A/Y registers are converted to floating point by CVTFPN. This value is then multiplied by 256 by loading the A/Y registers with the address of a 256 floating point constant (\$89,\$00,\$00,\$00,\$00) and then calling MULTAY. The result, in FPA1, is copied across to FPA2 (because it will be needed there later) and the low value byte is then read from memory. This value is also converted to floating point format in FPA1 by loading the A register with 0 and calling CVTFPN.

FPA1 and FPA2 can now be added together by calling the ADD routine (having first set the zero flag to reflect the exponent of FPA1). The result (low byte + 256*high byte) is now in FPA1 (which is where we want it) and the RTS instruction at the end of the ADD routine returns control to the Interpreter.

NEXT MONTH

Well, that's all there is to floating point numbers — and it's also all there is to this month's article. Next month I'll be looking at strings. I'll be showing you how to extract string parameters from program lines and how to use the string function routines. And of course, I'll be including some string-based keywords to illustrate the principles involved.


MICRO
CHOICE
OUR COMPUTER BUYER'S GUIDE
£2.25
Summer
1984
Advice and hints on
how to choose
your micro
ver 25 in-depth reviews
f microcomputers and
eripherals including:
tari 600XL, BBC Micro,
ommodore 64,
harp MZ-700, Memotech,
3C Micro, Electron
rinters and more
ORIC
OUT NOW — ALL THE FACTS
N COMPUTING HARDWARE
preview

BACKNUMBERS

MARCH 1983
Colour Genie reviewed, Epson HX-20 review, PEEKing the Spectrum, Into Atari's BASIC, Terminology translated.

APRIL 1983
Froglet on the BBC Micro, PC-1251 hand-held review, Valley Variations, Galaxy reviewed, Micro Database, Lower case UK101.

MAY 1983
Spectrum Book Survey, Oric-1 Review, Going FORTH Again, Jupiter Ace review.

JUNE 1983
Interrupt handling, Rubic simulation on the Spectrum, Beating the RS232 Blues, Lynx review, Indexer.

JULY 1983
Atari renumber, 16-bit micros, Bomb-proof Tandy, Olivetti Praxis 30 review, Ikon Hobbit tape drive review.

AUGUST 1983
Speeding up the Sharp, Premier Dragon disc drive, Sord M5 review, BBC String Store, Planetfall.

SEPTEMBER 1983
FELIX knowledge shops, Software protection, Torch disc pack, ZX81 Backgammon, Dragon character generator, Three Tandy computers.

OCTOBER 1983
Slingshot game, Sharp MZ-700 review, Sharp MZ-3541 review, Z80 Disassembler, A better TRSDOS, Improved VIC-20 editor.

NOVEMBER 1983
BBC Word Processor, ZX LPRINT review, Laser 200 review, Writing Adventures, Learning FORTH Part 1, PET tape append.

DECEMBER 1983
MIKRO assembler review, Getting More from the 64 Part 1, Adventures part 2, Curve-fitting, BBC Touch Typing Tutor.

JANUARY 1984
TRS-80 programmer's aid, Apple music, Electron review, TRS-80 screen editor, calendar program.

FEBRUARY 1984
Using MX-80 graphics, Colour Genie monitor, non-random random numbers, ZX81-FORTH, Program recovery on the Commodore 64.

MARCH 1984
Easycode part 1, BBC poker, Spectrum SCOPE review, Genie utilities, Spectrum Centronics interface.

APRIL 1984
Memotech MTX500 review, Genie BASIC extensions, Brainstorm review, Disassembly techniques, Recursion.

If you've lost, lent or had stolen one of those precious back copies of Computing Today then now is your chance to fill the gap in your collection. The list of issues given here represents the few remaining copies that we have available to help complete your library of all that's good in features, programs and reviews.

If you want one of these issues, it's going to cost you £1.40 (including postage and packing) but we think that's a small price to pay for the satisfaction you'll get. Ordering could hardly be made simpler — just fill in the form, cut it out (or send a photocopy) together with your money to:

**Backnumbers,
Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.**

If you wait until next month to do it, the chances are that we'll have run out of the very issue you wanted!

A Computing today Reader Service

BACKNUMBERS

Please send me the following Backnumbers

ISSUE

At £1.40 each. I enclose £ ....

NAME ....

ADDRESS ....

....

....

.... POSTCODE ....

Signature ....

I enclose a cheque/PO for £ ...... (Payable to ASP Ltd)
I wish to pay by credit card

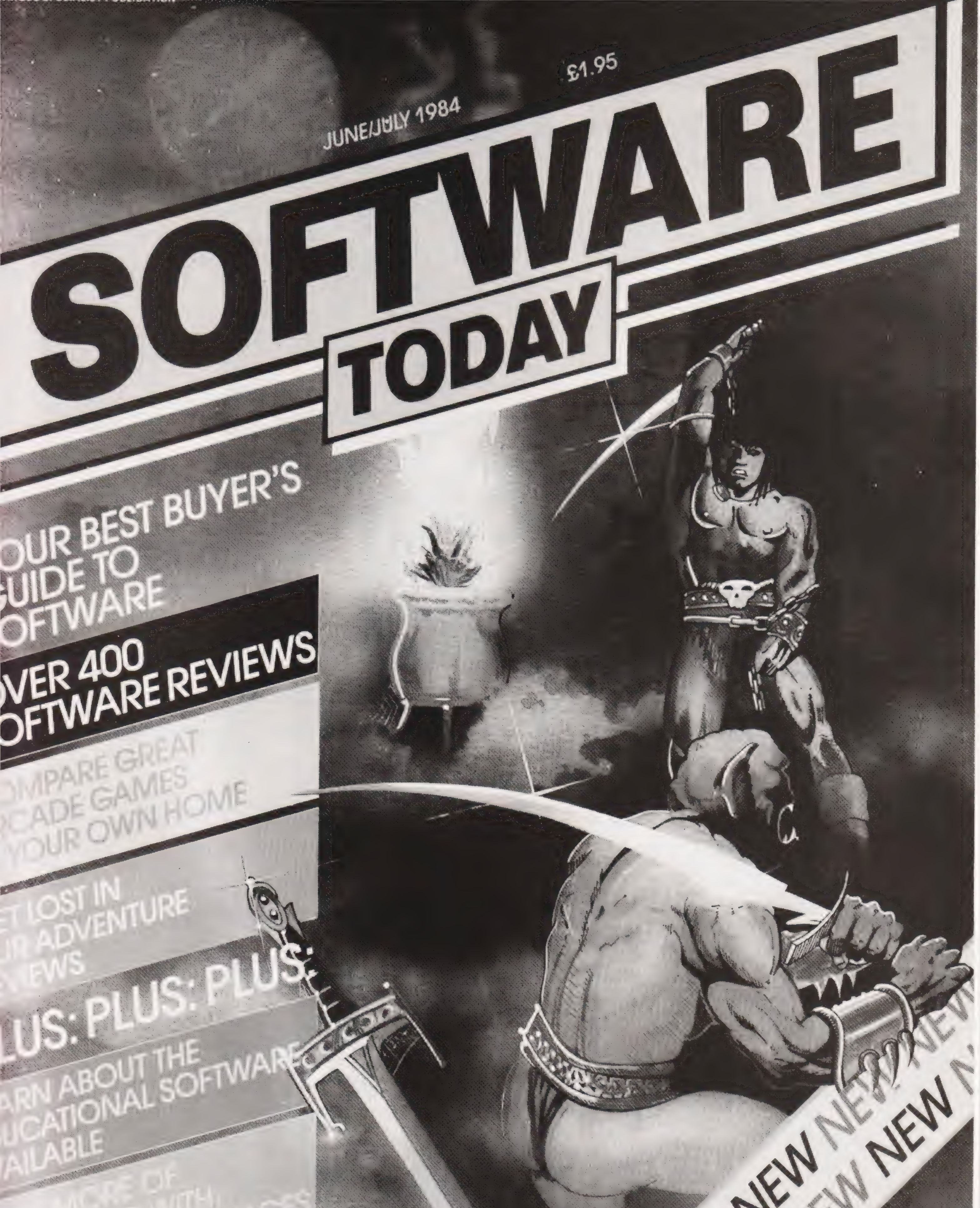
Access ☐ Barclaycard ☐

Insert Card No.

If you wish to pay by Access or Barclaycard, just fill in your card number and sign the form, do not send your card.

Please allow 21 days for delivery.

CT July '84


£1.95
JUNE/JULY 1984
SOFTWARE
TODAY
OUR BEST BUYER'S
UIDE TO
OFTWARE
VER 400
OFTWARE REVIEWS
OMPARE GREAT
CADE GAMES
YOUR OWN HOME
T LOST IN
R ADVENTURE
EWS
LUS: PLUS: PLUS:
ARN ABOUT THE
UCATIONAL SOFTWARE
AILABLE
NEW NEW NEW NEW
NEW NEW NEW

THE ART OF ISLAM

G.W. Gallagher

Discard your ruler and compass and create intricate geometric patterns on the BBC or Apple with a program that gives a fascinating insight into the graphic capabilities of both machines.

When in the seventh century, Islam spread into countries which had been part of the Byzantine Empire, scholars became acquainted with the geometrical concepts which played such an important part in the relationship of the universe to mathematical form and symmetry.

The use of ruler and compasses is the basis for the designs which occur in Islamic art and architecture, whether the buildings are over a 1000 years old or twentieth century. Designs are based on regular polygons placed to form regular or sem-regular grids. A regular grid uses only one basic polygon, such as a square or a hexagon, wheras a semi-regular grid uses more than one type of polygon, such as an octagon combined with a square. Polygons may be placed side by side, overlapping or overlaid to form the most complex patterns

The original programs illustrated in this article were written for a BBC Model B but as all my previous pattern and symmetry programs had been based on an Apple, adjustments have been made so that the same effects may be achieved on the Apple. The main adjustments needed are given at the end of the article.

Starting each time with a circle, the type of pattern is determined by the number of equal arcs into which the circumference is divided. A good draughtsman (with time and patience) can produce an infinite variety of designs but why not use a computer to carry out all the repetitive work?

HEXAGON BASED DESIGNS

When the circumference of a circle is divided into six equal parts, the outline of a regular hexagon is produced. If the circle is drawn on graph paper, the coordinates of useful points may be picked out and used as required (calculate them if you prefer to do so!)

For example, using a circle of radius 100 units and a centre with coordinates (XC, YC), two different hexagons are produced (see Figures 1 and 2). Considering the BBC computer (for Apple see later) the following procedures print the appropriate hexagon when given the centre of the shape, each forming a regular grid (Figure 3).

PROCHEX	(2000-2080) Uses the hexagon of Figure 1
PROCHEX2	(1000-1080) Uses the hexagon of Figure 2
PROCC2	(2200-2280) Concentrating first of all on the patterns which may be based on Figure 1, it is useful to be able to call on a 'filled-in' hexagon of a specified colour and another procedure is added.
PROCC1	(2110-2220) When the hexagons are placed side by side as shown in Figure 4, a diamond shaped space is left between them. PROCC1 will colour in such a diamond (Figures 4 and 5).
PROCC4	(2500-2570) If alternate vertices of the hexagon are joined, making two equilateral triangles, the effect is of a six-pointed star. This procedure colours in the star (Figure 6).

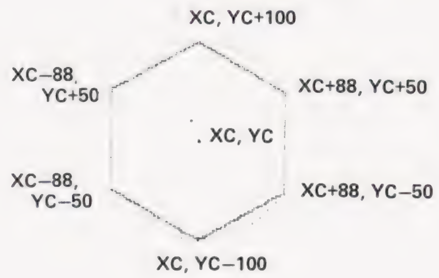


Fig. 1

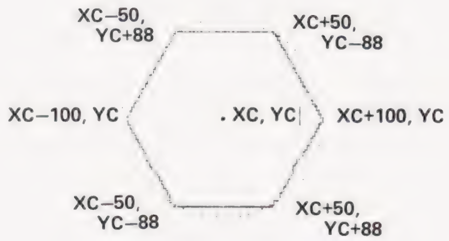


Fig. 2

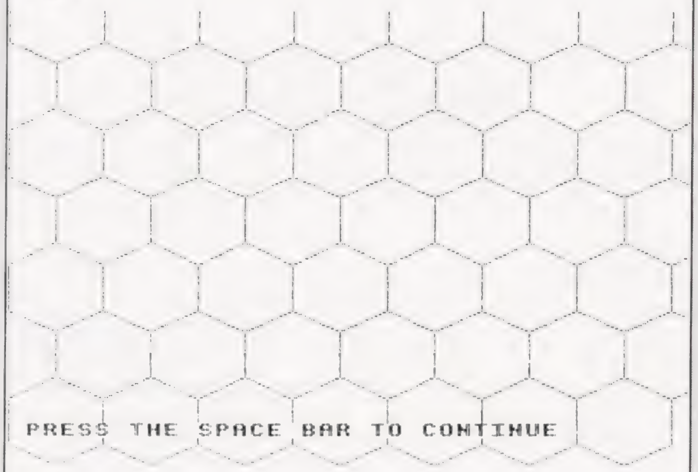


Fig. 3

Fig. 4

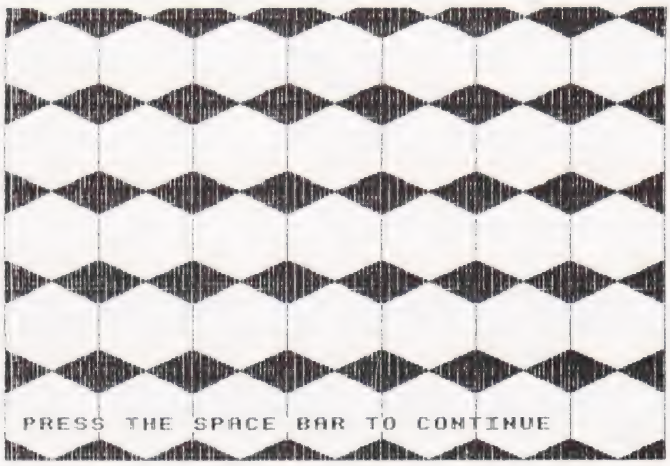


Fig. 5

```
10 MODE1
20 VDU19.2.6.0.0.0
30 VDU19.1.7.0.0.0
100 FORQ=0TO5
110   CX=88:CY=100
120   FORP=-1TO6
130     XC=CX+176*P
140     YC=CY+300*Q
150     PROCHEX
160     NEXTP
170   CX=176:CY=250
180   FORP=-1TO6
190     XC=CX+176*P
200     YC=CY+300*Q
210     PROCHEX
220     NEXTP
230   NEXTQ
240 PROCWAIT:CLS:K=1
250 FORQ=0TO5
260   CX=88:CY=100
270   FORP=-1TO6
280     XC=CX+176*P
290     YC=CY+200*Q
300     GOSUB 3100
310     NEXTP
320   NEXTQ
330 PROCWAIT
350 K=K+1:IFK>4 THEN 360 ELSE 250
360 R=2:K=1:CLS
370 FORQ=-1 TO 7
380   CX=88:CY=50*R
390   FORP=-1 TO 6
400     XC=CX+R*176*P
410     YC=CY+R*100*Q
420     GOSUB 3300
430     NEXTP
440   CX=88+88*R:CY=100*R
450   FORP=-1TO6
460     XC=CX+R*176*P
470     YC=CY+R*100*Q
480     GOSUB3300
490     NEXTP
500   NEXTQ
510 PROCWAIT:K=K+1:ON K GOTO 520,520,520,525,530,540,550,560,570,580,590,600,6
10
520 R=2:GOTO 370
525 CLS:R=2:GOTO 370
530 CLS:R=5/3:GOTO370
540 R=5/3:GOTO370
550  CLS:R=1.5:GOTO370
560 R=1.5:GOTO370
570 R=1.5:GOTO370
580 CLS:R=4/3:GOTO370
590 R=4/3:GOTO370
600 R=4/3:GOTO370
610 CLS
620 FORQ=-1TO6
630   CX=100:CY=100
640   FORP=-1TO6
650     XC=CX+300*P
660     YC=CY+176*Q
670     PROCHEX2
680     NEXTP
690   CX=250:CY=188
700   FORP=-1TO6
710      XC=CX+300*P
720     YC=CY+176*Q
730     PROCHEX2
740     NEXTP
750   NEXTQ
760 PROCWAIT:CLS
770 K=1
780 FORQ=-1TO6
790   CX=100:CY=200
800   FORP=-1TO6
810     XC=CX+200*P
820     YC=CY+352*Q
830     GOSUB 3400
840     NEXTP
850   CX=200:CY=376
860   FORP=-1TO6
870     XC=CX+200*P
880     YC=CY+352*Q
890     GOSUB3400
900     NEXTP
910   NEXTQ
920 PROCWAIT
930 K=K+1:ON K GOTO 940,950,940,950,950,990
940 CLS:GOTO 780
950 GOTO 780
990 END
1000 DEFPROCHEX2
1010 MOVEXC+100.YC
1020 DRAWXC+50.YC+88
1030 DRAWXC-50.YC+88
1040 DRAWXC-100.YC
1050 DRAWXC-50.YC-88
1060 DRAWXC+50.YC-88
1070 DRAWXC+100.YC
1080 ENDPROC
1100 DEFPROCC6
1110 MOVEXC.YC-176
1120 MOVEXC-50.YC-88
1130 PLOT85.XC+50.YC-88
1140 MOVEXC+100.YC
1150 PLOT85,XC+150.YC-88
1160 ENDPROC
1200 DEFPROCC7
1210 MOVEXC+100.YC
1220 DRAWXC-50,YC+88
1230 DRAWXC-50,YC-88
1240 DRAWXC+100,YC
1250 MOVEXC+50,YC+88
1260 DRAWXC-100.YC
1270 DRAWXC+50.YC-88
1280 DRAWXC+50.YC+88
1290 ENDPROC
1300 DEFPROCC8
1310 MOVEXC+100.YC
1320 MOVEXC-50.YC+88
1330 PLOT85.XC-50.YC-88
1340 MOVEXC+50.YC+ 88
1350 MOVEXC-100.YC
1360 PLOT85,XC+50,YC-88
1370 ENDPROC
2000 DEFPROCHEX
2010 MOVEXC+88.YC+50
2020 DRAWXC,YC+100
2030 DRAWXC-88.YC+50
2040 DRAWXC-88.YC-50
2050 DRAWXC,YC-100
2060 DRAWXC+88,YC-50
2070 DRAWXC+88,YC+50
2080 ENDPROC
2100 DEFPROCC1
2110 MOVEXC-88.YC-150
2120 MOVEXC-88,YC-50
2130 PLOT85.XC,YC-100
2140 MOVEXC+88,YC-50
2150 PLOT85,XC+88.YC-150
2160 ENDPROC
2200 DEFPROCC2
2210 MOVEXC-88.YC+50
2220 MOVEXC-88.YC-50
2230 PLOT85.XC,YC+100
2240 PLOT85.XC.YC-100
2250 PLOT85.XC+88.YC-50
2260 MOVEXC,YC+100
2270 PLOT85.XC+88.YC+50
2280 ENDPROC
2400 DEFPROCC3
2410 MOVEXC+44.YC-75
2420 MOVEXC+44,YC+75
2430 PLOT85,XC-88,YC
2440 MOVEXC-44,YC-75
2450 MOVEXC+88,YC
2460 PLOT85.XC-44.YC+75
2470 ENDPROC
2500 DEFPROCC4
2510 MOVEXC.YC+100
2520 MOVEXC+88,YC-50
2530 PLOT85.XC-88,YC-50
2540 MOVEXC+88.YC+50
2550 MOVEXC-88.YC+50
2560 PLOT85.XC.YC-100
2570 ENDPROC
2600 DEFPROCC5
2610 MOVEXC+30.YC-48
2620 MOVEXC+30.YC+48
2630 PLOT85.XC-60.YC
2640 MOVEXC-30.YC-48
2650 MOVEXC+60.YC
2660 PLOT85.XC-30.YC+48
2670 ENDPROC
3000  DEFPROCWAIT
3010  PRINTTAB(1,29);"PRESS THE SPACE BAR TO CONTINUE"
3020 G$=GET$:IFG$="" THEN 3020
3030  ENDPROC
3100 IFK=1 THEN PROCHEX:GOTO3140
3110 IFK=2 THEN GCOLO.2:PROCC1:GOTO3140
3120 IFK=3 THEN GCOLO.1:PROCC2:GOTO3140
3130 IFK=4 THEN GCOLO.0:PROCC4
3140 RETURN
3300 ON K GOTO 3310,3320,3325,3325,3310,3320,3310,3320,3330,3310,3320,3335
3310 GCOLO.3:PROCHEX:GOTO 3340
3320 GCOL1,2:PROCC2:GOTO 3340
3325  GCOLO.1:PROCC4:GOTO 3340
3330 GCOLO.0:PROCC3:GOTO 3340
3335  GCOLO.1:PROCC5
3340 RETURN
3400 IFK=1 THEN PROCHEX2:GOTO 3490
3410 IFK=2 THEN GCOLO.2:PROCC6:GOTO 3490
3420 IFK=3 THEN PROCC7:GOTO 3490
3430 IFK=4 THEN GCOLO.2:PROCC8:GOTO 3490
3440 IFK=5 THEN GCOLO.3:PROCC6
3490 RETURN
```

Listing 1. The hexagon program for the BBC.

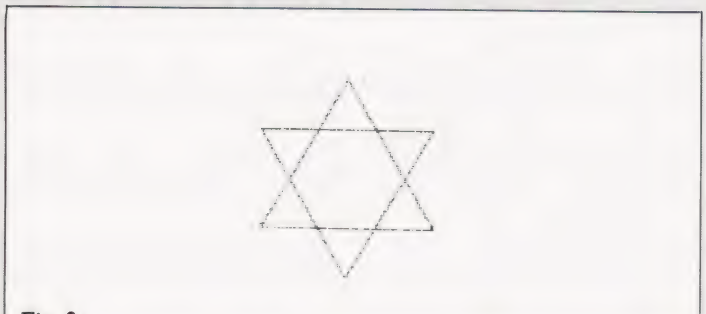


Fig. 6

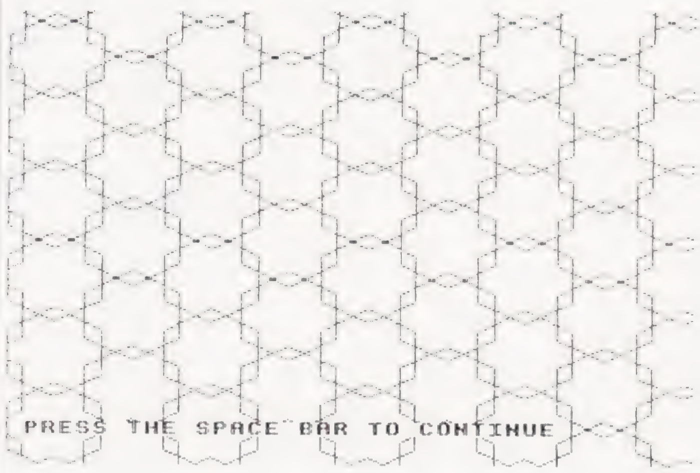


Fig. 8

OVERLAPPING THE HEXAGONS

If we start from the network of hexagons shown in Figure 7, the distance between adjacent x-coordinates of the centres of the hexagons is twice the radius of the circumscribing circle. The same is true of the y-coordinate of the centres. When the distance between the centres is less than twice the radius, the hexagons overlap.

Using R as a multiplying factor, different examples of overlapping patterns are obtained, Figures 8, 9 and 10 show the results when R = 5/3, 3/2 and 4/3 respectively. Combine the values of R with different procedures and an increased variety of patterns results. The only colours used in the program are black, white and blue-green which is CYAN (colour 6) on the BBC and HCOLOR 6 on the Apple.

Throughout the program, K is used as a counting factor to define the colour and procedure to be used and to decide whether to overlay a design on what is already there or to start with a clear screen. The maximum value of the variable P determines the number of repeats across the screen and Q the number of repeats vertically. The BBC is tolerant of values outside the screen size but, for the Apple, a check is needed to ensure the points are on the screen, otherwise an ERROR will be produced.

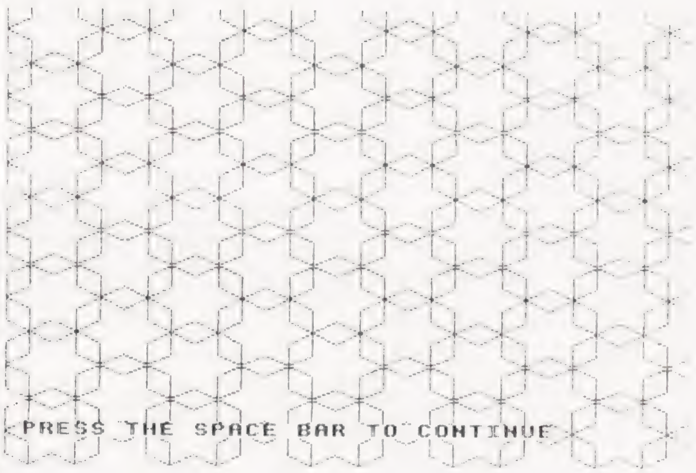


Fig. 9

THE FIRST SECTION

This uses the hexagon of Figure 1 followed by the coloured diamond and star. Figure 11 shows the result when the hexagon is white, the star black and the diamond cyan.

10 - 350 Main section of program.
3000-3030 PROCWAIT to move on to next pattern.
3100-3140 Subroutine to allocate colours and procedures.

THE SECOND SECTION

This uses overlapping hexagons, Figures 12 and 13.

360 - 610 Main section
3300-3340 Subroutine to allocate values of K and procedures.

When the overlapping hexagons are traced out on the screen, there are two other stars to be seen, PROCC3 and PROCC5 are

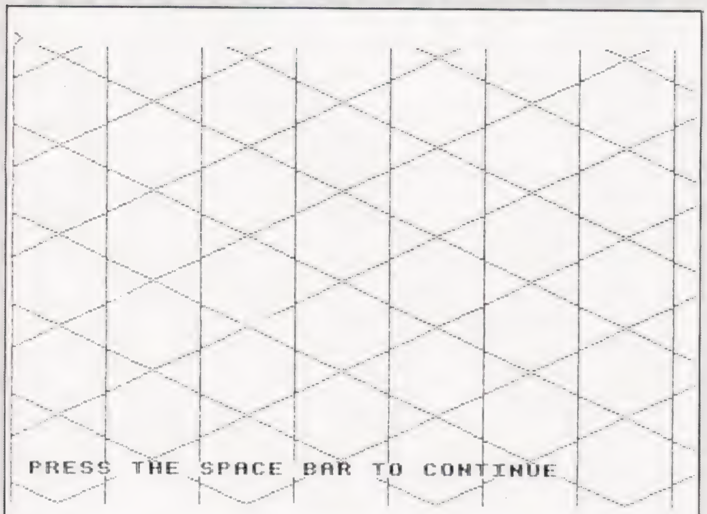


Fig. 7

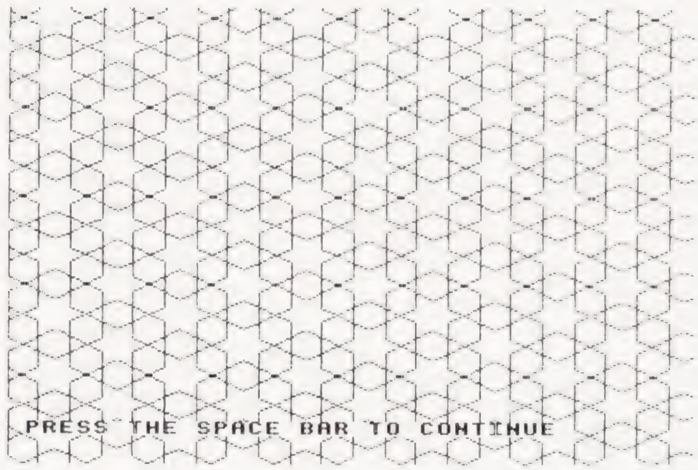


Fig. 10

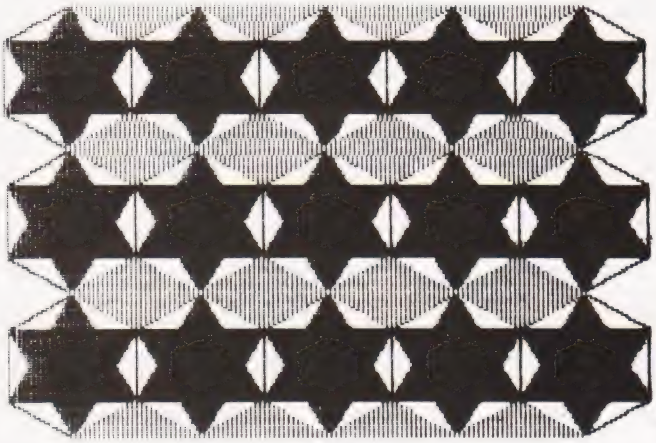


Fig. 11

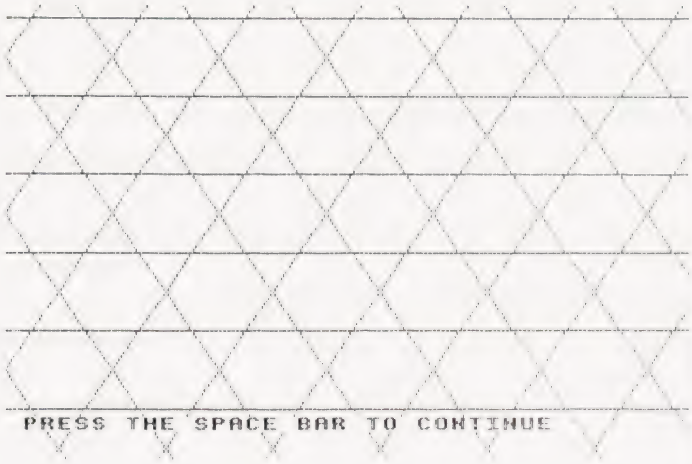


Fig. 14

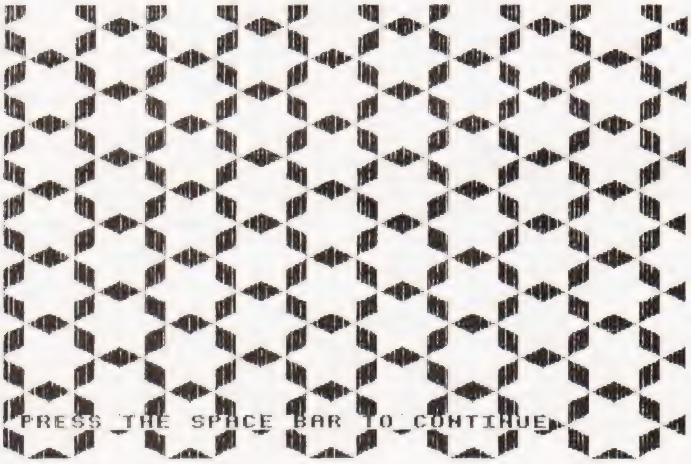


Fig. 12

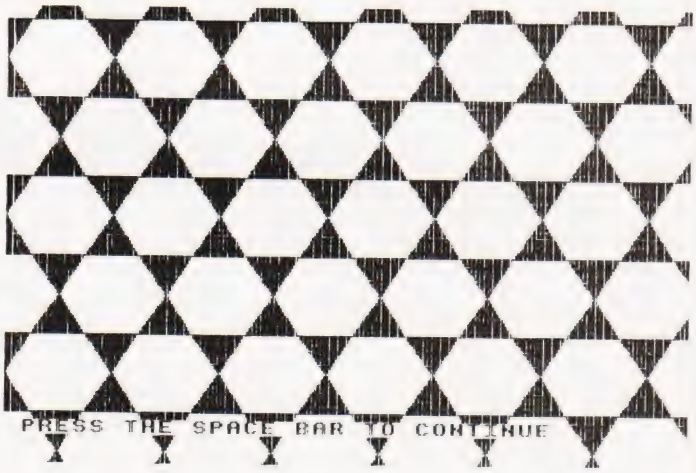


Fig. 15

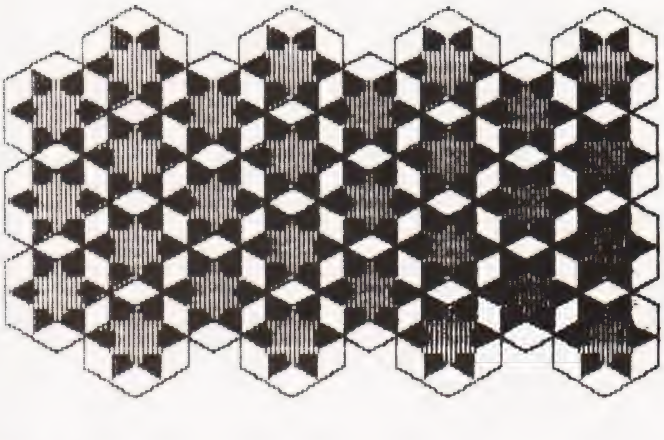


Fig. 13

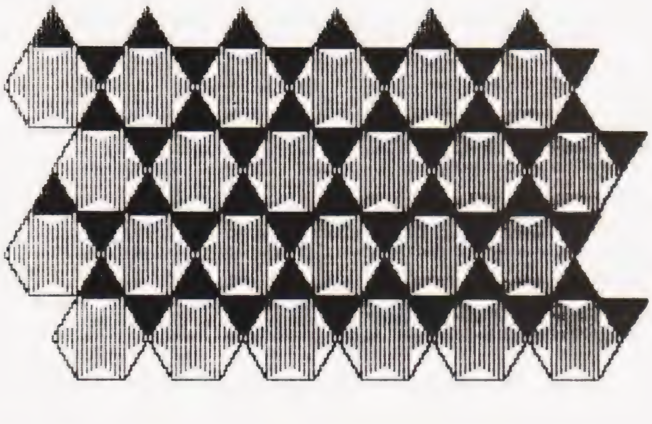


Fig. 16

included to fill in these stars.

PROCC3 (2400-2470)
PROCC5 (2600-2670)

THE THIRD SECTION

This uses the hexagon of Figure 2.

620 - 990 Main section.
3400-3490 Subroutine to allocate colour and shape.

In completing the plane patterns using hexagon 2, a larger star and a set of equilateral triangles may be filled in (PROCC6 and PROCC8 while PROCC7 provides a line version of the star.

PROCC6 (1100-1160) The larger star.
PROCC8 (1300-1370) Equilateral triangles.
PROCC7 (1200-1290) Line version of the star.

See Figures 14, 15, 16 and 17.

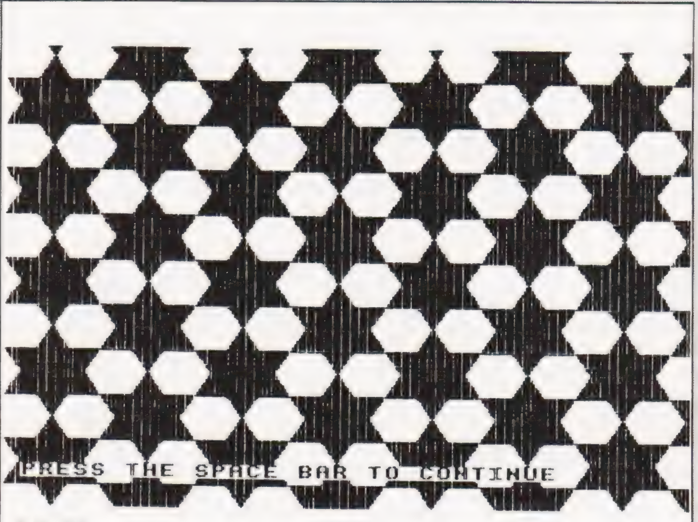


Fig. 17

```
10 MODE1
20 VDU19,2,6,0,0,0
30 REM....BBC...OCTAGON
100 K=1:R=2
110 FORQ=-1 TO6
120   CX=100:CY=100
130   FORP=-1 TO 8
140     XC=CX+100*P*R
150     YC=CY+100*Q*R
160     GOSUB3000
170     NEXTP
180   NEXTQ
190 PROCWAIT
200 K=K+1:ON K GOTO 210,210,220,230,240,250,260,280
210 R=2:GOTO 110
220 R=2:GOTO 110
230 CLS:R=1.4:GOTO 110
240 R=1.4:GOTO 110
250 CLS:R=2:GOTO 110
260 R=2:GOTO 110
280 K=1:CLS
290 FORQ=-1TO6
300   CX=100:CY=100
310   FORP=-1 TO 8
320     XC=CX+200*P
330     YC=CY+200*Q
340     GOSUB3500
350     NEXTP
360   NEXTQ
370 PROCWAIT
380 K=K+1:IFK=6 THEN 410 ELSE 390
390 IFK=4 THEN 400 ELSE 290
400  CLS:GOTO 290
410  CLS:K=1
420  CX=100:CY=100
430  FORQ=-1 TO 6
440    FORP=-1 TO 6
450      XC=CX+200*P
460      YC=CY+200*Q
470     GOSUB3600
480     NEXTP
490   NEXTQ
500 K=K+1:ON K GOTO 510,510,520,530,990
510 CX=200:CY=200:GOTO 430
520  PROCWAIT:CX=100:CY=100:GOTO 430
530  CX=100:CY=100:GOTO430
990 END
2000 DEFPROCOCT1
2010 MOVEXC+100,YC
2020 DRAWXC+70,YC+70
2030 DRAWXC,YC+100
2040 DRAWXC-70,YC+70
2050 DRAWXC-100,YC
2060 DRAWXC-70,YC-70
2070 DRAWXC,YC-100
2080 DRAWXC+70,YC-70
2090 DRAWXC+100,YC
2100 ENDPROC
2110 DEFPROCC1
2120 MOVEXC+70,YC+70
2130 MOVEXC+100,YC
2140 PLOT85,XC,YC+100
2150 PLOT85,XC-70,YC+70
2160 MOVEXC+100,YC
2170  PLOT85,XC-100,YC
2180  PLOT85,XC+70,YC-70
2190  PLOT85,XC,YC-100
2200 MOVEXC-100,YC
2210  PLOT85,XC-70,YC-70
2220 ENDPROC
2230 DEFPROCC2
2240 MOVEXC+70,YC+70
2250 MOVEXC,YC
2260 PLOT85,XC,YC+40
2270 PLOT85,XC-70,YC+70
2280 MOVEXC,YC
2290 PLOT85,XC-40,YC
2300 PLOT85,XC-70,YC-70
2310 MOVEXC,YC
2320 PLOT85,XC,YC-40
2330 PLOT85,XC+70,YC-70
2340 MOVEXC,YC
2350 PLOT85,XC+40,YC
2360 PLOT85,XC+70,YC+70
2370 ENDPROC
2400 DEFPROCC3
2410 MOVEXC+70,YC+70
2420 MOVEXC-70,YC+70
2430 PLOT85,XC+70,YC-70
2440 PLOT85,XC-70,YC-70
2450 MOVEXC+100,YC
2460 MOVEXC,YC+100
2470 PLOT85,XC,YC-100
2480 PLOT85,XC-100,YC
2490 ENDPROC
2500 DEFPROCOCT2
2510 MOVEXC+100,YC+41
2520 DRAWXC+41,YC+100
2530 DRAWXC-41,YC+100
2540 DRAWXC-100,YC+41
2550 DRAWXC-100,YC-41
2560 DRAWXC-41,YC-100
2570 DRAWXC+41,YC-100
2580 DRAWXC+100,YC-41
2590 DRAWXC+100,YC+41
2600 ENDPROC
2610 DEFPROCC4
2620 MOVEXC+41,YC+100
2630 MOVEXC+100,YC+41
2640 PLOT85,XC-41,YC+100
2650 PLOT85,XC-100,YC+41
2660 MOVEXC+100,YC+41
2670 PLOT85,XC-100,YC-41
2680 PLOT85,XC+100,YC-41
2690 PLOT85,XC-41,YC-100
2700 PLOT85,XC+41,YC-100
2710 ENDPROC
2720 DEFPROCC6
2730 MOVEXC+70,YC-70
2740 MOVEXC+130,YC-70
2750 PLOT85,XC+70,YC-130
2760 PLOT85,XC+130,YC-130
2770 MOVEXC+100,YC-55
2780 MOVEXC+55,YC-100
2790 PLOT85,XC+145,YC-100
2800 PLOT85,XC+100,YC-145
2810 ENDPROC
2900 DEFPROCWAIT
2910 PRINTTAB(1,29);"PRESS THE SPACE KEY TO CONTINUE"
2920 IF GET$="" THEN 2920
2930 ENDPROC
3000 IFK=1 THEN PROCOCT1:GOTO 3070
3010 IFK=2 THEN GCOLO,2:PROCC1:GOTO 3070
3020 IFK=3 THEN GCOLO,3:PROCC2:GOTO 3070
3030 IFK=4 THEN GCOLO,3:PROCOCT1:GOTO 3070
3040  IFK=5 THEN GCOLO,2:PROCC2:GOTO 3070
3050  IFK=6 THEN GCOLO,2:PROCC1:GOTO 3070
3060  IFK=7 THEN GCOLO,3:PROCC3:GOTO 3070
3070 RETURN
3100 DEFPROCC5
3110 MOVEXC+70,YC-30
3120 MOVEXC+30,YC-30
3130 PLOT85,XC+70,YC-70
3140 PLOT85,XC+30,YC-70
3150 MOVEXC-70,YC-30
3160 MOVEXC-30,YC-30
3170 PLOT85,XC-70,YC-70
3180 PLOT85,XC-30,YC-70
3190 MOVEXC-70,YC+70
3200 MOVEXC-30,YC+70
3210 PLOT85,XC-70,YC+30
3220 PLOT85,XC-30,YC+30
3230 MOVEXC+70,YC+70
3240 MOVEXC+30,YC+70
3250 PLOT85,XC+70,YC+30
3260 PLOT85,XC+30,YC+30
3270 MOVEXC,YC-100
3280 MOVEXC-30,YC-70
3290 PLOT85,XC+30,YC-70
3300 PLOT85,XC,YC-40
3310 MOVEXC-40,YC
3320 MOVEXC-70,YC-30
3330 PLOT85,XC-70,YC+30
3340 PLOT85,XC-100,YC
3350 MOVEXC-30,YC+70
3360 MOVEXC,YC+40
3370  PLOT85,XC,YC+100
3380  PLOT85,XC+30,YC+70
3390  MOVEXC+70,YC+30
3400  MOVEXC+40,YC
3410  PLOT85,XC+100,YC
3420  PLOT85,XC+70,YC-30
3430 ENDPROC
3500 IFK=1 THEN PROCOCT2:GOTO 3580
3510 IFK=2 THEN GCOLO,3:PROCC4:GOTO 3580
3520 IFK=3 THEN GCOLO,2:PROCC3:GOTO 3580
3530 IFK=4 THEN GCOLO,2:PROCC5:GOTO 3580
3540 IFK=5 THEN GCOLO,3:PROCC6
3580 RETURN
3600 IFK=1 THEN PROCOCT2:GOTO 3650
3610 IFK=2 THEN PROCOCT2:GOTO 3650
3620 IFK=3 THEN GCOLO,2:PROCC4:GOTO 3650
3630 IFK=4 THEN GCOLO,0:PROCC7:GOTO 3650
3650 RETURN
3700 DEFPROCC7
3710 MOVEXC,YC+50
3720 MOVEXC+50,YC
3730 PLOT85,XC-50,YC
3740 PLOT85,XC,YC-50
3750 ENDPROC
```

Listing 2. The octagon program for the BBC.

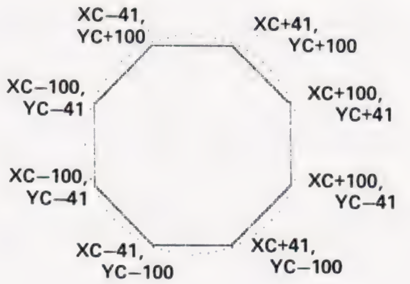


Fig. 18

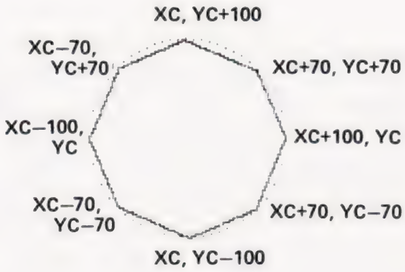


Fig. 19

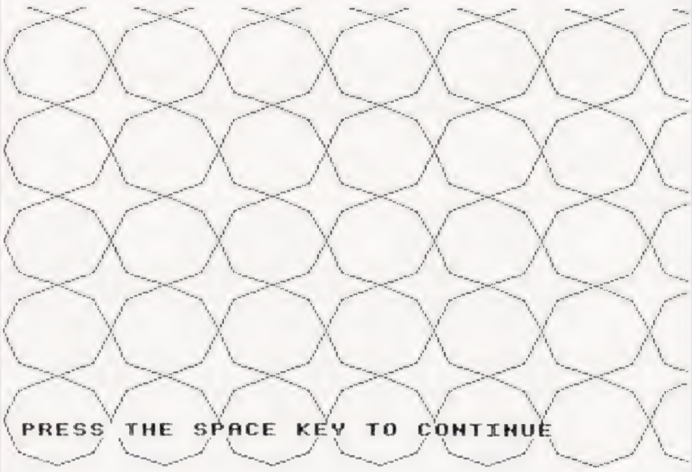


Fig. 20

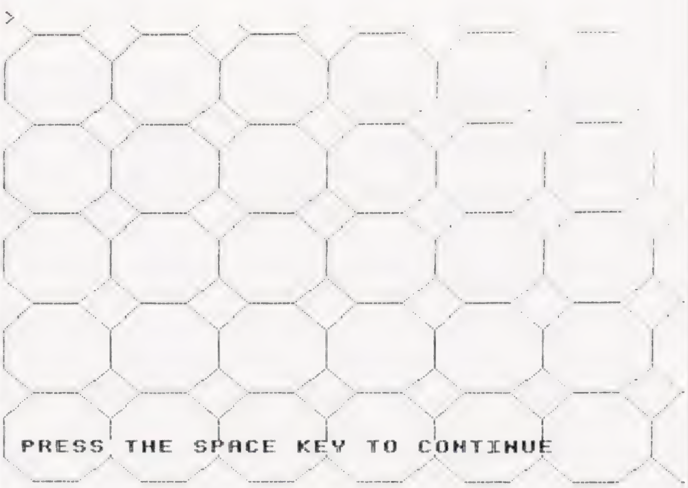


Fig. 21

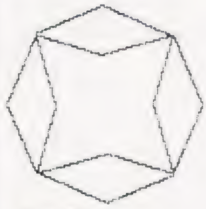


Fig. 22

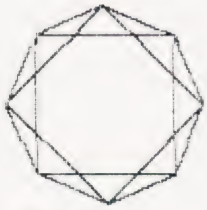


Fig. 23

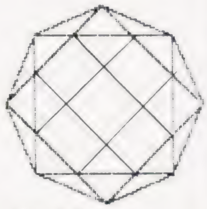


Fig. 24

OCTAGON BASED PATTERNS

When the circumference of the basic circle is divided into eight equal parts, the regular octagon is the result; when its vertices are picked out from an accurate drawing on squared paper (or calculated) Figures 18 and 19 are obtained.

Repeated octagons will not form a regular grid but two semi-regular grids may be drawn (Figures 20 and 21). As with the hexagon, the octagons are drawn using coordinates of vertices relative to a centre (XC, YC).

PROCOCT1	(2000-2100) draws the octagon in Figure 18.
PROCC1	(2110-2220) fills in the octagon with solid colour.
PROCC2	(2230-2370) fills in the four-pointed star which is formed by completing a rhombus on each pair of adjacent sides (Figure 22).
PROCC3	(2400-2490) When alternate points are joined to form two squares, the squares taken together form an eightpointed star; see Figure 23.
PROCOCT2	(2500-2600) The octagon in Figure 19 is outlined.
PROCC4	(2610-2710) Fills in the octagon in Figure 19.
PROCC7	(3700-3750) The square which is seen in the grid in Figure 21 is coloured, using the coordinate of the centre of the octagon which joins the upper left hand side of the square as reference.
PROCC6	(2720-2810) Fills in a smaller star of the same shape as PROCC3, in the position of the last square.
PROCC5	(3100-3430) Using the octagon in Figure 18; drawing extra lines (Figure 24) will produce a series of small squares enclosing another star.

The various procedures can be used to overlap or overlay shapes using either grid. A series of such combinations are joined together in the main program, using the variables K, R, P and Q as with the hexagon.

10 - 280	The main program using the octagon of Figure 18.
2900-2930	PROCWAIT
3000-3070	The subroutine to allocate colours and shapes in the first section.
290 - 530	The main program using the octagon of Figure 19.

3500-3580 The subroutine to allocate colours and shapes in the second section.
3600-3650 The subroutine for the more complex pattern.

There are many more complicated patterns which may be transcribed for the computer screen, the three stages of Figure 34 give an example of the type of pattern which can be achieved.

These programs were written with the aim of making them easily adaptable for other types of computer but some of the other graphic capabilities of the BBC computer can be used with considerable effect. Using other GCOL commands in the subroutines which allocate colour and shape and using PLOT 77 to fill in appropriate areas instead of relying on combinations of triangles can also offer different constructions.

VARIATIONS FOR THE APPLE

The program can be adjusted for any computer which has a high resolution graphics screen. The following adjustments necessary for the Apple may be useful as a guide.

- The screen size has to be adjusted from 1280 by 1024 to 280 by 160 or 192.
- The MOVE and DRAW commands have to be replaced by HPLOT.
- The PLOT 85 command has been replaced by a subroutine at 2100 in both programs which fills in a triangle between any three given points. Since the routine uses a divisor (X2−X1) the first and second points should not have the same x coordinate.
- The PROCs have been replaced by subroutines.
- PROCWAIT has been replaced by a subroutine at 2900 in the octagon program which allows the pattern to be printed out; Figure 35 (see Apple Listings 1 and 2).

```
  10 REM...SAMPLE HEXAGON LISTING FOR APPLE
  90 HGR:HCOLOR=3
 100 K=1:R=2
 110 FOR Q=0 TO 3
 120   CX=22:CY=12.5*R
 130   FOR P=-1 TO 9
 140     XC=CX+44*P*R:IF XC>250 OR XC<21 THEN 170
 150     YC=CY+25*Q*R:IF YC>167 OR YC<25 THEN 170
 160     GOSUB3000
 170     NEXT P
 180   CX=22+22*R:CY=25*R
 185   FOR P=-1 TO 9
 190     XC=CX+44*P*R:IF XC>260 OR XC<21 THEN 220
 200     YC=CY+25*Q*R:IF YC>167 OR YC<25 THEN 220
 210     GOSUB3000
 220     NEXT P
 230   NEXT Q
 235 GOSUB2500
 240 K=K+1
 250 IF K=2 THEN HGR:R=5/3:GOTO 110
 260 IF K=3 THEN HGR:R=3/2:GOTO 110
 270 IF K=4 THEN HGR:R=4/3:GOTO 110
 280 IF K=5 THEN HGR:R=1:GOTO 110
 290 IF K=6 THEN HGR:R=2:GOTO 110
 300 IF K=7 THEN GOTO 110
 310 IF K=8 THEN GOTO 110
 320 IF K=9 THEN HGR:GOTO 110
 330 IF K=10 THEN HGR:R=3/2:GOTO 110
 340 IF K=11 THEN R=3/2:GOTO 110
 350 IF K=12 THEN R=3/2:GOTO 110
 990 END
1000 HPLOT XC+22,YC+12.5 TO XC,YC+25 TO XC-22,YC+12.5 TO XC-22,YC-12.5 TO XC,YC
25 TO XC+22,YC-12.5 TO XC+22,YC+12.5
1010 RETURN
1100 X1=XC+22:X2=XC-22:X3=XC:Y1=YC+12.5:Y2=YC+12.5:Y3=YC-25:GOSUB2100
1110 RETURN
1200 X1=XC:X2=XC-22:X3=XC+22:Y1=YC+25:Y2=YC-12.5:Y3=YC-12.5:GOSUB2100
1210 RETURN
1400  X1=XC+20:X2=XC-10:X3=XC-10:Y1=YC:Y2=YC+17:Y3=YC-17:GOSUB2100
1410  X1=XC+10:X2=XC-20:X3=XC+10:Y1=YC+17:Y2=YCY3=YC-17:GOSUB2100
1420  RETURN
1500  X1=XC+11:X2=XC-11:X3=XC:Y1=YC+6.25:Y2=YC+6.25:Y3=YC-12.5:GOSUB2100
1510  RETURN
1600  X1=XC:X2=XC-11:X3=XC+11:Y1=YC+12.5:Y2=YC-6.25:Y3=YC-6.25:GOSUB2100
1610  RETURN
2100 D=X1-X2:IF D>0 THEN A=-.5:GOTO 2120
2110  IF D<0 THEN A=.5:GOTO 2120
2115  IF D=0 THEN 2160
2120  FOR X=X1 TO X2 STEP A
2130   Y=Y1+(X-X1)*(Y2-Y1)/(X2-X1)
2140   HPLOT X3,Y3 TO X,Y
2150    NEXT X
2160  RETURN
2500 VTAB21:PRINT "P FOR PRINT,N NOT"
2510  INPUT P$:IF P$<>"P" THEN 2600
2520  PRINTCHR$(4);"PR£1"
2530  PRINTCHR$(17)
2540  PRINTCHR$(4);"PR£0"
2600  RETURN
3000  IF K<6 THEN GOSUB1000:GOTO 3090
3010  IF K=6 THEN GOSUB1000:GOTO 3090
3020  IF K=7 THEN GOSUB1100:GOTO 3090
3030  IF K=8 THEN GOSUB1200:GOTO 3090
3040  IF K=9 THEN HCOLOR=6:GOSUB1100:GOSUB1200:GOTO 3090
3050  IF K=10 THEN HCOLOR=3:GOSUB1000:GOTO 3090
3060  IF K=11 THEN HCOLOR=3:GOSUB1400:GOTO 3090
3070  IF K=12 THEN HCOLOR=6:GOSUB1500:GOSUB 1600
3090  RETURN
```

Listing 3. Sample Hexagon listing for the Apple.

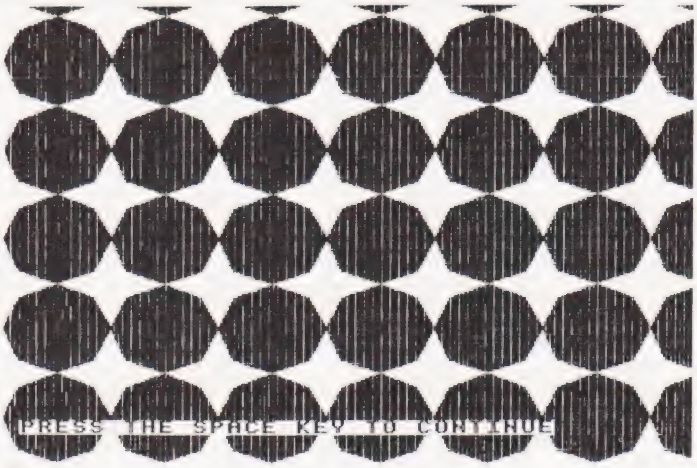


Fig. 25

Fig. 26

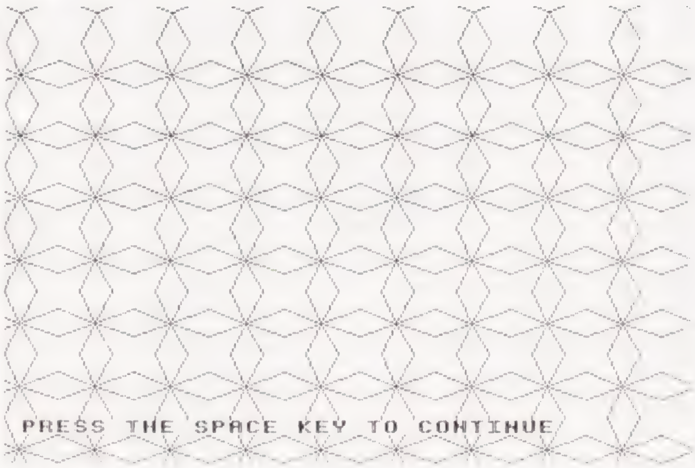


Fig. 27

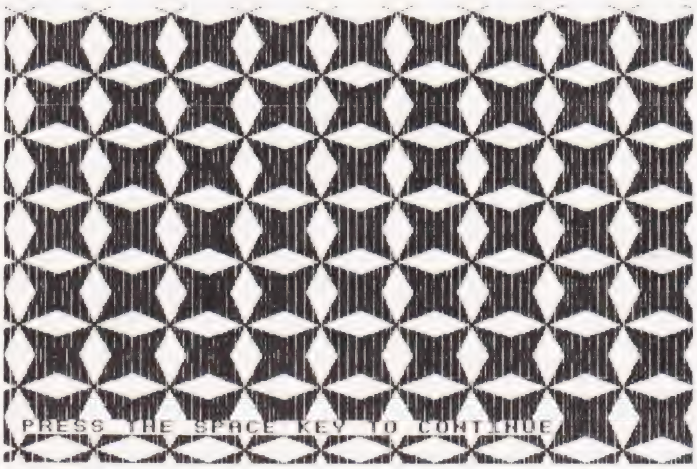


Fig. 28

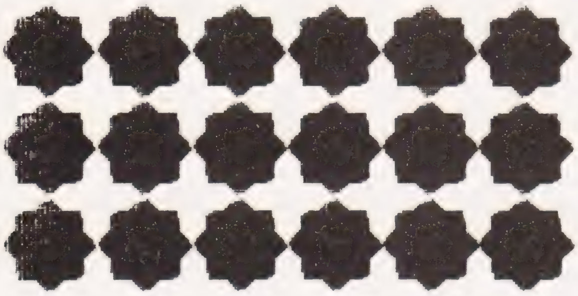


Fig. 29

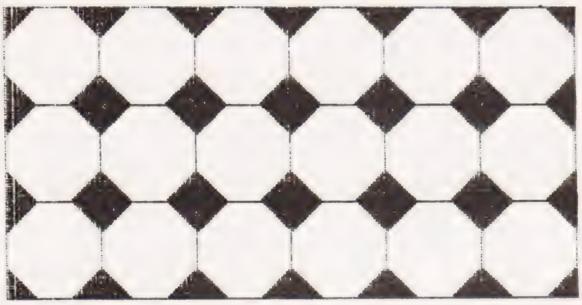


Fig. 30

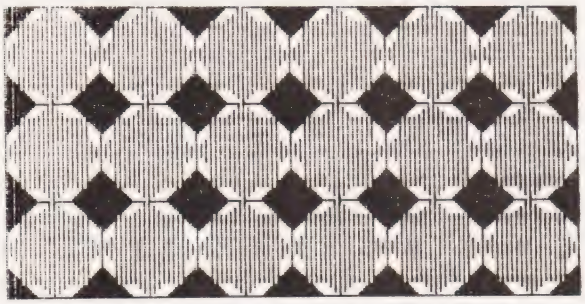


Fig. 31

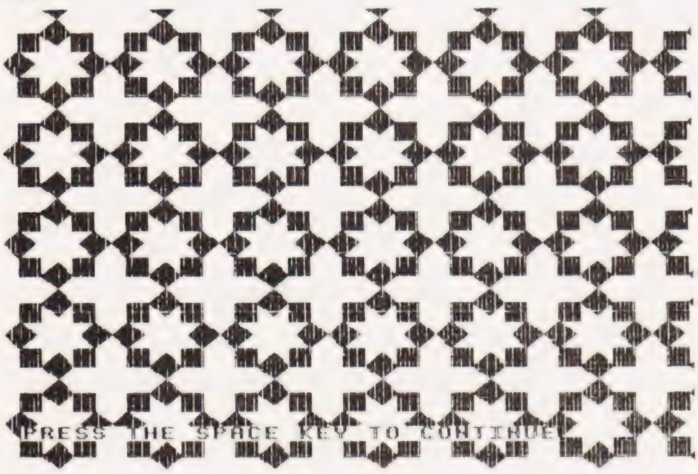


Fig. 32

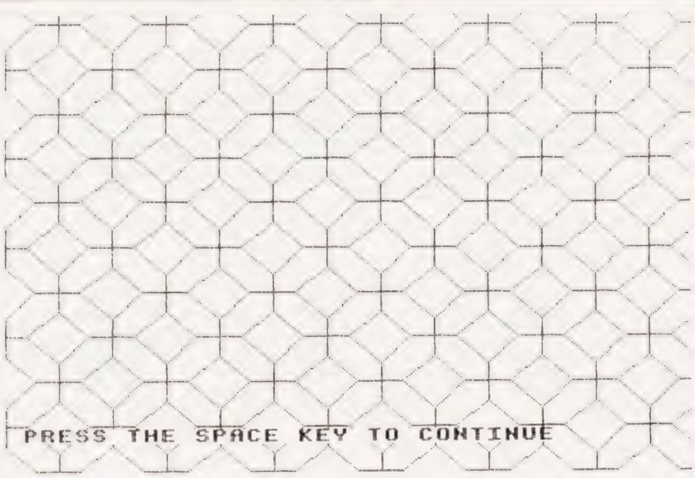
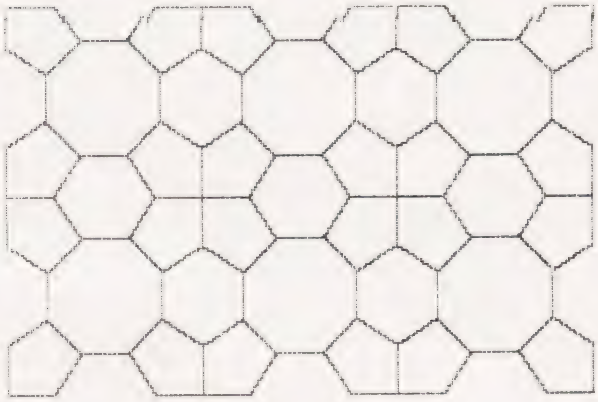
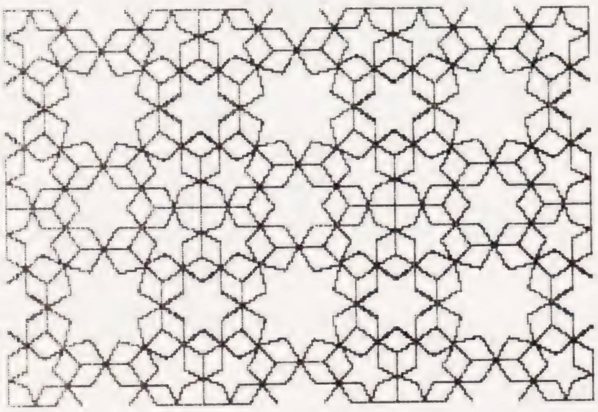


Fig. 33





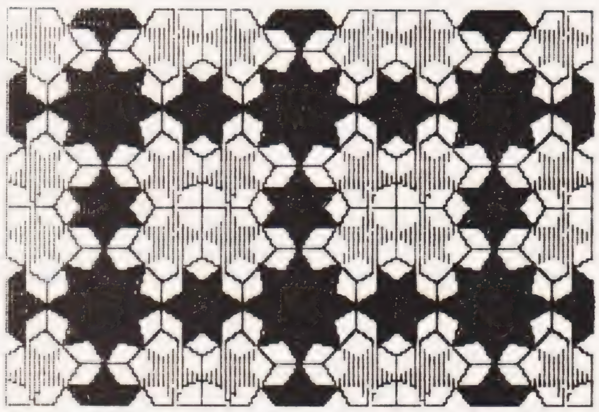


Fig. 34

```
 80 REM...APPLE LISTING FOR OCTAGON
 90 HGR:HCOLOR=3
 100 K=1:R=2
 110 FOR Q=0 TO 5
 120   CX=20:CY=20
 130   FOR P=0 TO 6
 140      XC=CX+20*P*R:IF XC>259 OR XC<20 THEN 170
 150     YC=CY+20*Q*R:IF YC>139 OR YC<20 THEN 170
 160     GOSUB 3000
 170     NEXT P
 180   NEXT Q
 190 GOSUB 2900
 200 K=K+1:ON K GOTO 210,210,220,230,240,250,260,270,280
 210 R=2:GOTO 110
 220 R=2:GOTO 110
 230 HOME:HGR:R=1.4:GOTO 110
 240 R=1.4:GOTO 110
 250 HOME:HGR:R=2:GOTO 110
 260 R=2:GOTO 110
 270 R=2:GOTO 110
 280 HGR:HCOLOR=3:K=1
 290 FOR Q=0 TO 5
 300   CX=20:CY=20
 310    FOR P = 0 TO 8
 320     XC=CX+40*P:IF XC>259 OR XC<20 THEN 350
 330     YC=CY+40*Q:IF YC>139 OR YC<20 THEN 350
 340     GOSUB 3500
 350     NEXT P
 360   NEXT Q
 370 GOSUB2900
 380 K=K+1:IF K=5 THEN 410
 390 IF K=4 THEN HGR:GOTO 290
 400 GOTO 290
 410 HGR:HCOLOR=3:K=1
 420 CX=20:CY=20
 430 FOR Q=0 TO 5
 440   FOR P=0 TO 6
 450     XC=CX+40*P:IF XC>259 OR XC<20 THEN 480
 460     YC=CY+40*Q:IF YC>139 OR YC<20 THEN 480
 470     GOSUB 3600
 480     NEXT P
 490   NEXT Q
 500 K=K+1:ON K GOTO 510,510,520,530,540,550
 510 CX=40:CY=40:GOTO 430
 520 GOSUB 2900:CX=20:CY=20:GOTO 430
 530 CX=20:CY=20:GOTO 430
 540 GOTO 990
 550 GOTO 990
 990  END
2000 HPLOT XC+20,YC TO XC+14,YC+14 TO XC,YC+20 TO XC-14,YC+14 TO XC-20,YC
2010 HPLOT TO  XC-14,YC-14 TO XC,YC-20 TO XC+14,YC-14 TO XC+20,YC
2020 RETURN
2100 D=X1-X2:IF D>0 THEN A=-.5:GOTO 2120
2110 IF D<0 THEN A=.5:GOTO 2120
2120 FOR X=X1 TO X2 STEP A
2130   Y=Y1+(X-X1)*(Y2-Y1)/(X2-X1)
2140   HPLOT X3,Y3 TO X,Y
2150   NEXT X
2160 RETURN
2170 X1=XC+20:Y1=YC:X2=XC+14:Y2=YC+14:X3=XC:Y3=YC:GOSUB 2100
2200 X1=XC+20:Y1=YC:X2=XC+14:Y2=YC+14:X3=XC:Y3=YC:GOSUB 2100
2210  X1=XC+14:Y1=YC+14:X2=XC:Y2=YC+20:X3=XC:Y3=YC:GOSUB 2100
2220  X1=XC:Y1=YC+20:X2=XC-14:Y2=YC+14:X3=XC:Y3=YC:GOSUB 2100
2230  X1=XC-14:Y1=YC+14:X2=XC-20:Y2=YC:X3=XC:Y3=YC:GOSUB 2100
2240  X1=XC-20:Y1=YC:X2=XC-14:Y2=YC+14:X3=XC:Y3=YC:GOSUB 2100
2250  X1=XC-14:Y1=YC-14:X2=XC:Y2=YC-20:X3=XC:Y3=YC:GOSUB 2100
2270  X1=XC:Y1=YC-20:X2=XC+14:Y2=YC-14:X3=XC:Y3=YC:GOSUB 2100
2280  X1=XC+14:Y1=YC-14:X2=XC+20:Y2=YC:X3=XC:Y3=YC:GOSUB 2100
2290  RETURN
2300  X1=XC+8:Y1=YC:X2=XC+14:Y2=YC+14:X3=XC:Y3=YC:GOSUB 2100
2310  X1=XC+14:Y1=YC+14:X2=XC:Y2=YC+8:X3=XC:Y3=YC:GOSUB 2100
2320  X1=XC:Y1=YC+8:X2=XC-14:Y2=YC+14:X3=XC:Y3=YC:GOSUB 2100
2330  X1=XC-14:Y1=YC+14:X2=XC-8:Y2=YC:X3=XC:Y3=YC:GOSUB 2100
2340  X1=XC-8:Y1=YC:X2=XC-14:Y2=YC-14:X3=XC:Y3=YC:GOSUB 2100
2350  X1=XC-14:Y1=YC-14:X2=XC:Y2=YC-8:X3=XC:Y3=YC:GOSUB 2100
2360  X1=XC:Y1=YC-8:X2=XC+14:Y2=YC-14:X3=XC:Y3=YC:GOSUB 2100
2370  X1=XC+14:Y1=YC-14:X2=XC+8:Y2=YC:X3=XC:Y3=YC:GOSUB 2100
2380  RETURN
2400  X1=XC+14:Y1=YC+14:X2=XC+8:Y2=YC:X3=XC+20:Y3=YC:GOSUB 2100
2410  X1=XC+14:Y1=YC-14:X2=XC+8:Y2=YC:X3=XC+20:Y3=YC:GOSUB 2100
2420  X1=XC+14:Y1=YC-14:X2=XC:Y2=YC-8:X3=XC:Y3=YC-20:GOSUB 2100
2430  X1=XC-14:Y1=YC-14:X2=XC:Y2=YC-8:X3=XC:Y3=YC-20:GOSUB 2100
2440  X1=XC-14:Y1=YC-14:X2=XC-20:Y2=YC:X3=XC-8:Y3=YC:GOSUB 2100
2450  X1=XC-14:Y1=YC+14:X2=XC-20:Y2=YC:X3=XC-8:Y3=YC:GOSUB 2100
2480  RETURN
2500  X1=XC-20:Y1=YC:X2=XC:Y2=YC+20:X3=XC:Y3=YC:GOSUB 2100
2510  X1=XC+20:Y1=YC:X2=XC:Y2=YC+20:X3=XC:Y3=YC:GOSUB 2100
2520  X1=XC+20:Y1=YC:X2=XC:Y2=YC-20:X3=XC:Y3=YC:GOSUB 2100
2530  X1=XC-20:Y1=YC:X2=XC:Y2=YC-20:X3=XC:Y3=YC:GOSUB 2100
2540  X1=XC-14:Y1=YC+14:X2=XC+14:Y2=YC+14:X3=XC:Y3=YC:GOSUB 2100
2550  X1=XC+14:Y1=YC-14:X2=XC+14:Y2=YC+14:X3=XC:Y3=YC:GOSUB 2100
2560  X1=XC+14:Y1=YC-14:X2=XC-14:Y2=YC-14:X3=XC:Y3=YC:GOSUB 2100
2570  X1=XC-14:Y1=YC+14:X2=XC-14:Y2=YC-14:X3=XC:Y3=YC:GOSUB 2100
2580  RETURN
2700 HPLOTXC+20,YC+8 TO XC+8,YC+20 TO XC-8,YC+20 TO XC-20,YC+8 TO XC-20,YC-8
2710 HPLOT TO XC-8,YC-20 TO XC+8,YC-20 TO XC+20,YC-8 TO XC+20,YC+8
2720 RETURN
2800  X1=XC-20:Y1=YC+20:X2=XC-8:Y2=YC+20:X3=XC-20:Y3=YC+8:GOSUB 2100
2810  X1=XC+20:Y1=YC+20:X2=XC+8:Y2=YC+20:X3=XC+20:Y3=YC+8:GOSUB 2100
2820  X1=XC+20:Y1=YC-8:X2=XC+8:Y2=YC-20:X3=XC+20:Y3=YC-20:GOSUB 2100
2830  X1=XC-20:Y1=YC-8:X2=XC-8:Y2=YC-20:X3=XC-20:Y3=YC-20:GOSUB 2100
2840  RETURN
2900  VTAB21:PRINT "PRESS 'P' TO PRINT, 'N' NOT"
2910  INPUT P$:IF P$ <> "P" THEN 2960
2930  PRINT CHR$(4);"PR£1"
2940  PRINT CHR$(17)
2950  PRINT CHR$(4);"PR£0"
2960  RETURN
3000  IF K=1 THEN GOSUB 2000:GOTO 3070
3010  IF K=2 THEN GOSUB 2200:GOTO 3070
3020  IF K=3 THEN HCOLOR=0: GOSUB 2300:GOTO 3070
3030  IF K=4 THEN HCOLOR=3: GOSUB 2000:GOTO 3070
3040  IF K=5THEN HCOLOR=3: GOSUB 2400:GOTO 3070
3050  IF K=6 THEN HCOLOR=3:GOSUB 2000:GOTO 3070
3060  IF K=7 THEN HCOLOR=3:GOSUB 2500:GOTO 3070
3065  IF K=8 THEN HCOLOR=0:GOSUB 2000
3070  RETURN
3100  FOR KX=-1 TO 1 STEP 2
3110    FOR KY=-1 TO 1 STEP 2
3120      X1=XC+KX*6:Y1=YC-KY*14:X2=XC:Y2=YC-KY*20:X3=XC:Y3=YC-KY*8:GOSUB2100
3130      X1=XC+KX*14:Y1=YC-KY*14:X2=XC+KX*6:Y2=YC-KY*14:X3=XC+KX*6:Y3=YC-KY*5:
GOSUB2100
3140      X1=XC+KX*6:Y1=YC-KY*6:X2=XC+KX*14:Y2=YC-KY*6:X3=XC+KX*14:Y3=YC-KY*14:
GOSUB2100
3150      X1=XC+KX*20:Y1=YC:X2=XC+KX*8:Y2=YC:X3=XC+KX*14:Y3=YC-KY*6:GOSUB2100
3160      NEXTKY
3170    NEXT KX
3180  RETURN
3400 X1=XC:Y1=YC+10:X2=XC+10:Y2=YC:X3=XC-10:Y3=YC:GOSUB2100
3410 X1=XC:Y1=YC-10:X2=XC+10:Y2=YC:X3=XC-10:Y3=YC:GOSUB2100
3430  RETURN
3500  IF K=1 THEN GOSUB2700:GOTO 3550
3510  IF K=2 THEN HCOLOR=3:GOSUB2800:GOTO 3550
3520  IF K=3 THEN HCOLOR=6:GOSUB2500:GOTO 3550
3530  IF K=4 THEN HCOLOR=3:GOSUB3100
3550  RETURN
3600  IF K=1 THEN HCOLOR=3:GOSUB2700:GOTO 3650
3610  IF K=2 THEN GOSUB 2700:GOTO 3650
3620  IF K=3 THEN HCOLOR=3:GOSUB 2800:GOTO 3650
3630  IF K=4 THEN HCOLOR=3:GOSUB 3400
3650  RETURN
```

Listing 4. Octagon listing on the Apple.

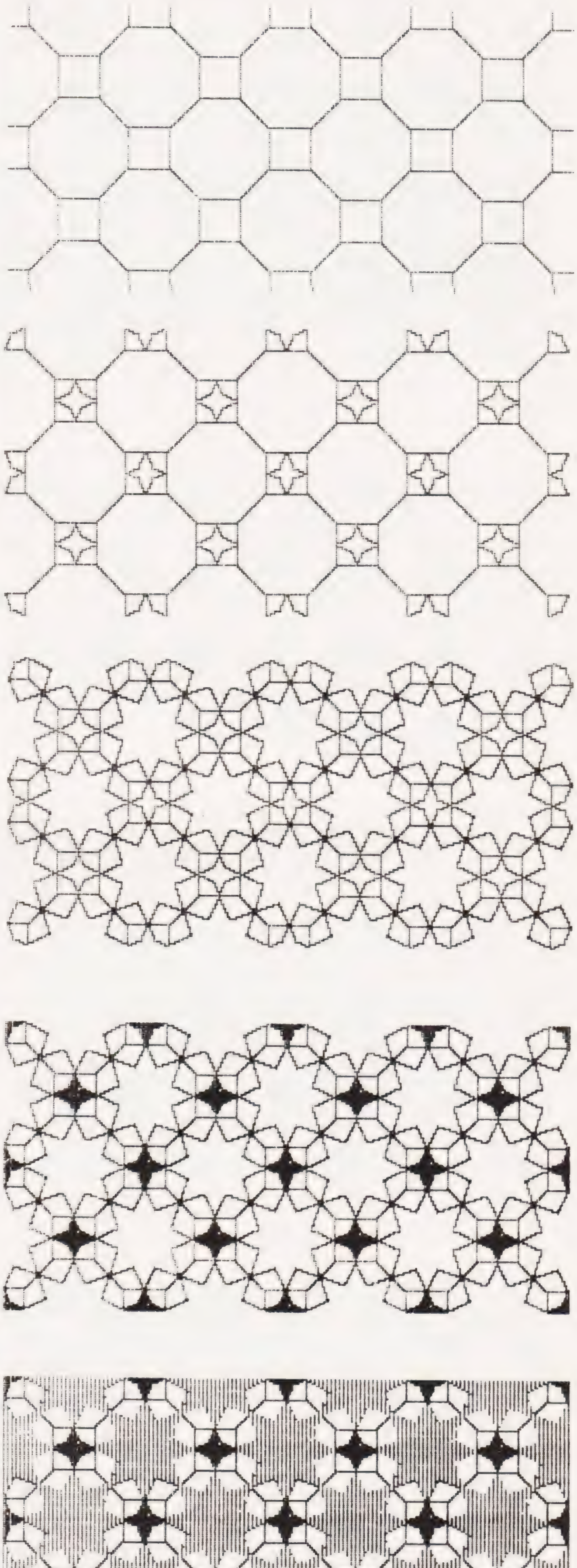


Fig. 35 Five stages in drawing an octagon pattern.

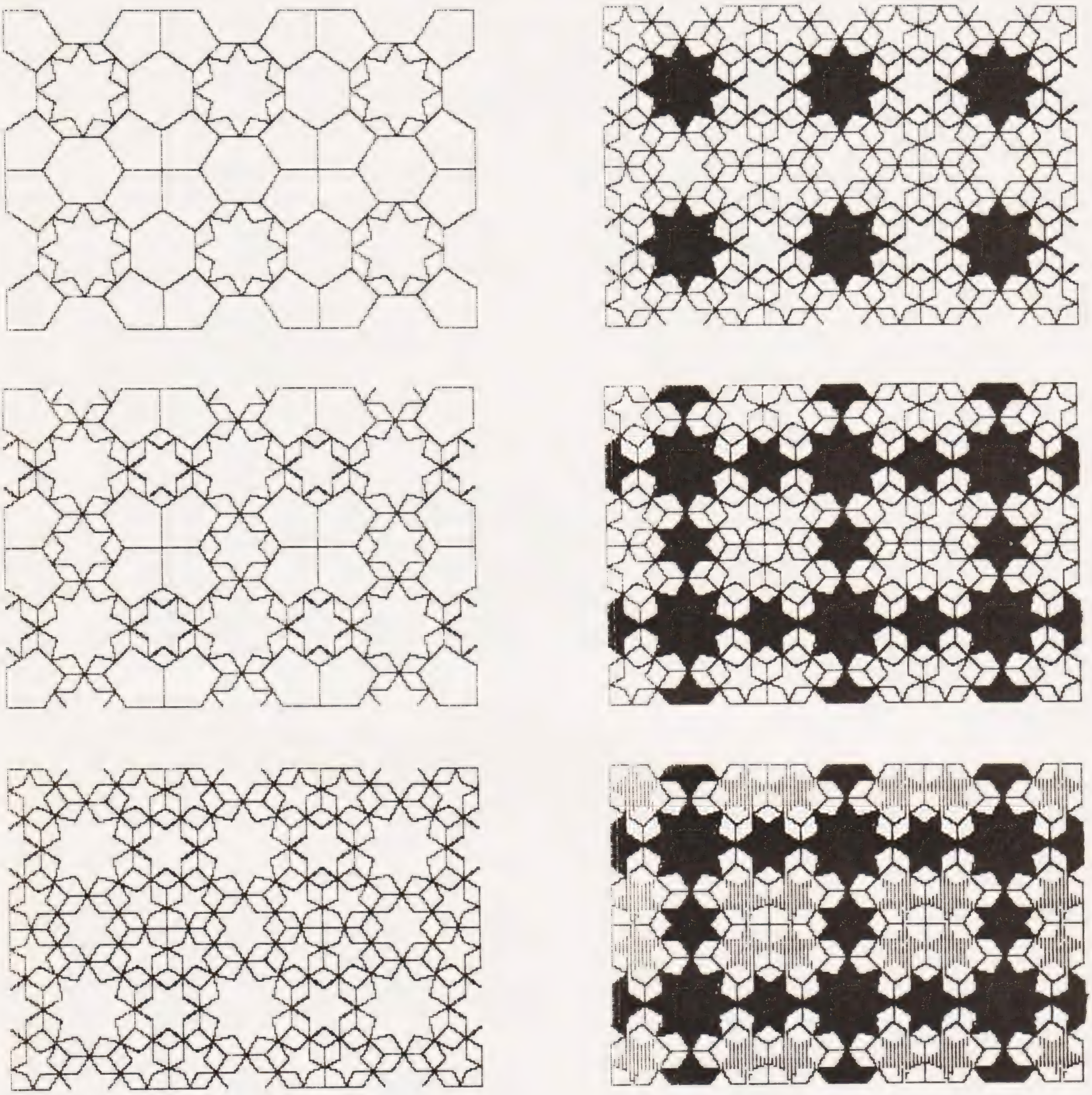


Fig. 36 Building up another complex shape with octagons.

Fig. 37

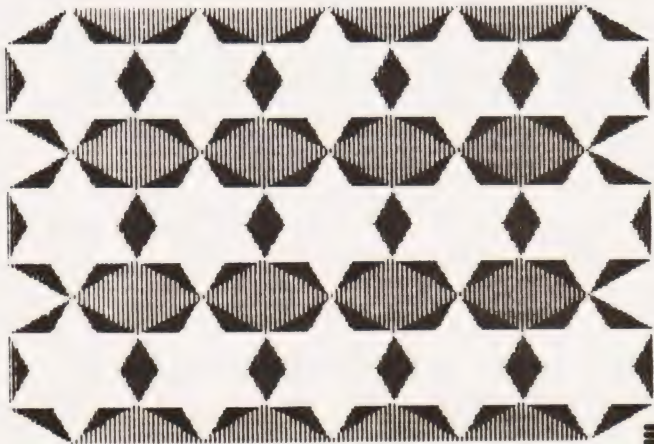


Fig. 38

ARCADE CHARTBUSTERS

BRITAIN'S SOFTWARE CHARTS

1	JET SET WILLY (Software Projects) — Spectrum	(1)
2	CHEQUERED FLAG (Psion) — Spectrum	(9)
3	MANIC MINER (Software Projects) — CBM 64	(—)
4	ATIC ATAC (Ultimate) — Spectrum	(3)
5	SCUBA DIVE (Durrell) — Spectrum	(2)
6	JACK & THE BEANSTALK (Thor) — Spectrum	(—)
7	CRAZY KONG (Intega Micros) — VIC 20	(—)
8	MOON ALERT (Ocean) — Spectrum	(—)
9	CODE NAME MAT (Micromega) — Spectrum	(7)
10	THE GUARDIAN (PSS) — Spectrum	(—)

NON-ARCADE CHARTBUSTERS

BRITAIN'S SOFTWARE CHARTS

1	FALL OF ROME (APS) — (Spectrum)	(—)
2	GOLF 64 (Abrasco) — CBM 64	(2)
3	FIGHTER PILOT (Digital) — Spectrum	(—)
4	FLIGHT PATH 737 (Anirog) — CBM 64	(—)
5	TWIN KINGDOM VALLEY (Bug Byte) — CBM 64	(1)
6	COLOSSUS CHESS TAPE (CDS) — CBM 64	(—)
7	STAR TRUCKER (Clever Clogs) — Spectrum	(—)
8	TEST MATCH (CRL) — Spectrum	(—)
9	THE WIZARD & THE PRINCESS (Melbourne House) — VIC 20	
10	POOL (Bug Byte) — Spectrum	(—)

by the ASP Market Research Group

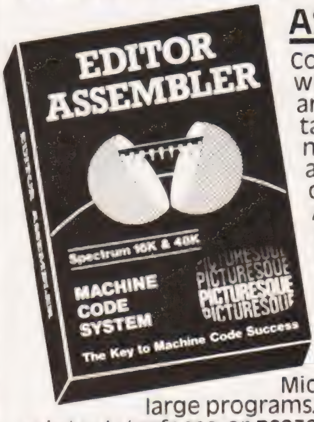
Compiled with the assistance of Britain's leading software distributors, including: Pinnacle, SDL, PCE, Websters, PCS and Software Centre.

The Key to Spectrum Machine Code Success.

Picturesque's MACHINE CODE SYSTEM is used and recommended by professional software writers, yet the excellent documentation and the friendly, easy-to-use programs have been highly recommended for beginners.

You will only buy one Machine Code System, so buy the best, the one the professionals use.

NEW IMPROVED PROGRAMS — MICRODRIVE COMPATIBLE



ASSEMBLER

Completely self-contained with its own line editor, giving an easy-to-read 40 column tabulated listing. Auto line numbering, line renumbering and auto tabulation make this one of the fastest and easiest Assemblers to use. 5-character label names. Decimal, Hex or ASCII constants. SAVE/LOAD/VERIFY both the listing and Machine Code to cassette/MICRODRIVE/NETWORK. Assemble from cassette or Microdrive into memory for very large programs. Customise to most Centronics printer Interfaces, or RS232 (with Interface 1) for 80 column printout. FAST ASSEMBLY — 1k of code in 7 seconds. Assembler Directives:— ORG, END, DEFB, DEFW, DEFM, EQU, DEFL. (Microdrive and Centronics facilities only operate with 48K machines.)

£8.50
INCL. VAT & P&P.

MONITOR

The ideal tool to help the beginner get started, yet it contains all the commands for the experienced programmer to run and de-bug machine code programs. Inspect and alter memory contents in Hex or ASCII. Breakpoints and full Register display. NOW WITH SINGLE STEPPING through RAM or ROM. Disassemble any part of memory, RAM or ROM. Dec-Hex-Dec number converter. Printer output to ZX printer or via RS232 (with Interface 1) or customise to most Centronics printer Interfaces. General memory management commands include Hex dump, Insert, Delete, Fill and Move. Can reside in memory with the Assembler (48K machines only) to give a complete Machine Code programming system.

£7.50
INCL. VAT & P&P.

Programs supplied on cassette with option to Save onto Microdrive (cartridge not supplied.)

Existing owners can obtain the new programs by returning the **cassette only** to Picturesque, along with a cheque/PO. for **£1.50 per program** (inc. VAT & P&P). New cassettes will be supplied by return of post

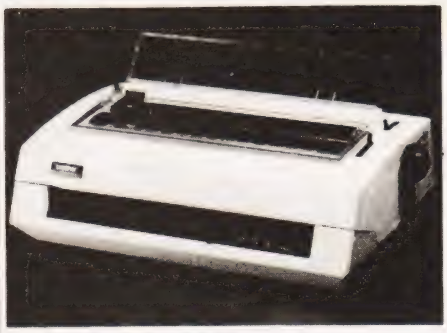
Available from the "SPECTRUM" chain of stores, branches of John Menzies and all good computer shops, or by mail order by sending cheque/PO to:
PICTURESQUE, 6 Corkscrew Hill, West Wickham, Kent, BR4 9BB. Send SAE for details.

PICTURESQUE

THE HR1

THE FINEST PRINTER/TYPEWRITERS MONEY CAN BUY.

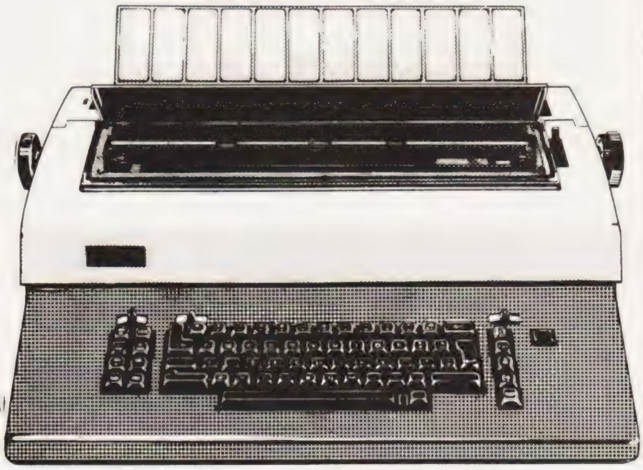
ONLY £595 + VAT



Compatible with
SHARP
OSBORNE
APPLE II
APPLE III
TELEVIDEO
TANDY
BBC
MIMI
TEXAS 30/30
PET
HEWLETT PACKARD
SUPERBRAIN
SIRIUS
XEROX
IBM
In fact most computers!

THE BROTHER CROWN RANIER

ONLY £695 + VAT



The HR1
Bi-directional – 17 CPS – some features as below, but without keyboard and lift off facility.

CROWN RANIER — The leader of all interfaced daisy wheel printers/typewriters.
Here's why.

The unbeatable Crown Ranier is the only heavy duty daisy wheel printer/typewriter which runs on a linear motor — no belts or cables to break, stretch of wear — and uses the IBM ribbon and lift off correctors. The reliability has become legend in these days of expensive electronic typewriter breakdowns.

CENTRONIC — IEEE — RS232 INTERFACES — Full KSR (keyboard send and receive). At small extra cost. No external boxes — interchangeable daisywheels — variable pitch — three free daisy wheels — automatic whole line correction — easily copes with standard printed forms — 135,000 characters per ribbon (at least twice the average character length) — standard lift off correctors — a printer/typewriter at the flick of a switch.

Perfection as a typewriter (now used by local authorities and multi-nationals), perfection as a printer.

Beware of lightweight, low cost machines — these can never give long term reliable service. THIS MACHINE WILL!!! The professional machine.

Ask your local computer or office equipment dealer for further information. EX-STOCK DELIVERY.

SOLE UK DISTRIBUTOR. TRADE ENQUIRIES INVITED

CROWN BUSINESS CENTRE

56-58 SOUTH ST., EASTBOURNE, SUSSEX. (0323) 639983.

BOOK PAGE

Garry Marshall

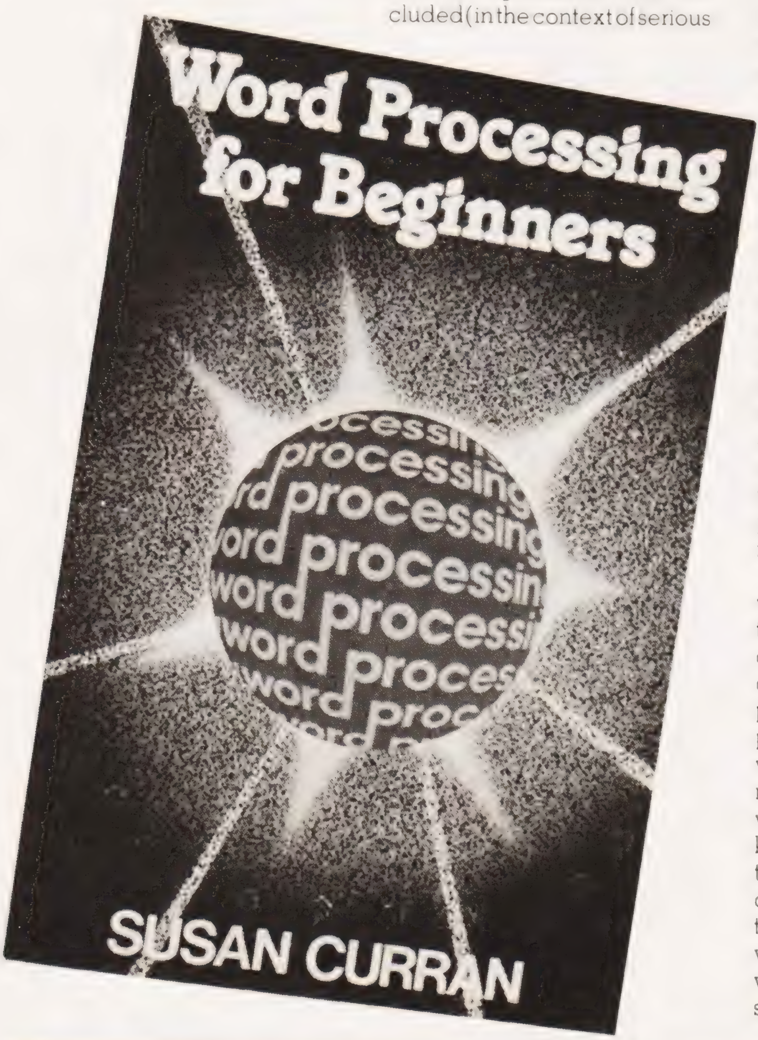
A pot-pourri of topics in our review page this month, with books on word processing, learning to program in BASIC and artificial intelligence on a micro.

This month's books are on word processing, first steps in BASIC and artificial intelligence. One provides a very good introduction to word processing, of the kind that I feel a great many people are looking for. There is a book that claims to be 'the easiest computer book on the market', a claim that I doubt and an aim that I find of doubtful value. The third book is also about word processing and, besides, can be seen as a great help to learning about computing, so that it complements the first two rather well. The final book is about how to make your Commodore 64 intelligent, and could make other computers intelligent, too, although not especially so.

Word Processing For Beginners by Susan Curran is a book that entirely lives up to its title. It begins with short and clear explanations of what word processing is and of what its benefits are when compared to using a typewriter. This is followed by an account of how you can assess your own word processing needs. If you want to write letters, your needs will clearly be different from those of the person who wants to use a word processor to write a book. There is a wide range of activities between these two extremes, and each has different requirements, brings different benefits and requires the selection and purchase of different equipment. All this is dealt with very purposefully, and the information and opinions given here will, alone, be worth the price of the book to many.

The author uses a word processor for writing books, and many other things, and her wealth of experience shows throughout the book in good, sound advice. I found a great deal with which I could only nod in agreement. This included (in the context of serious word processing, remember) calling the cheaper home computers toys by comparison with the more robustly constructed business computers, stressing the importance of an 80-column screen, explaining the definite advantages of a well-installed word processing program for a microcomputer when compared to a dedicated word processor, and stressing the importance of choosing your word processing program and then a computer that can run it rather choosing a computer first and proceeding from there.

These points and many others are fully expanded. I feel that it would be hard for the newcomer to word processing to find more sensible general advice than that given here.



There are two chapters each on word processing hardware and software. The hardware chapters cover computers and storage media, and then printers. In choosing a system, the choice of printer can be overlooked, but it is obviously vitally important (unless you intend to rely on electronic mail). Its price can be a significant factor, but it is also necessary to check that your computer can drive the printer you want to use, and a number of other things.

The software chapters deal with general matters first and then the particular features that a specific word processor may or may not possess. A chapter providing a glossary of word processing terms follows, providing a valuable source of reference. Terms like 'word wrap', 'justification' and 'marker' *do* need explaining. Each term is also accompanied by comments on who needs to use the feature, how it is done by a word processor and points to watch out for in the way that a specific word processor implements it. This is all very valuable.

The longest chapter, some 50 pages, describes some typical word processing programs, and uses test pieces to evaluate and compare their capabilities. Wordstar is here, along with a wide range of others from up-market ones to ones costing as little as £20. The book ends with short chapters on training in the use of word processors, setting up a system and producing standard letters.

I cannot praise the book too highly. It is lively, well written and clear. It is also a fund of useful information and sound advice.

Writing Your First Programs by John Parry is the book that claims to be 'the easiest computer book on the market'. I suppose that, if you can measure easiness, there has to be an easiest computer book. But that is not to say that it will be an easy book, because the ideas underlying computing and the relationships between them are not easy ones. This book is written as an 'easy book', and that seems to me to be its basic mistake (should it cause a syntax error?). The attempt gives rise to sentences such as 'You may have a small computer looking like a calculator with a window like the numbers on a calculator to see the answers.' A window like a number? And this is in the very first paragraph. This failing is compounded, for the author soon forgets that he is supposed to be writing an 'easy' book, and even as early as Chapter 2 starts to use 'difficult' four-syllable words such as 'complicated'.

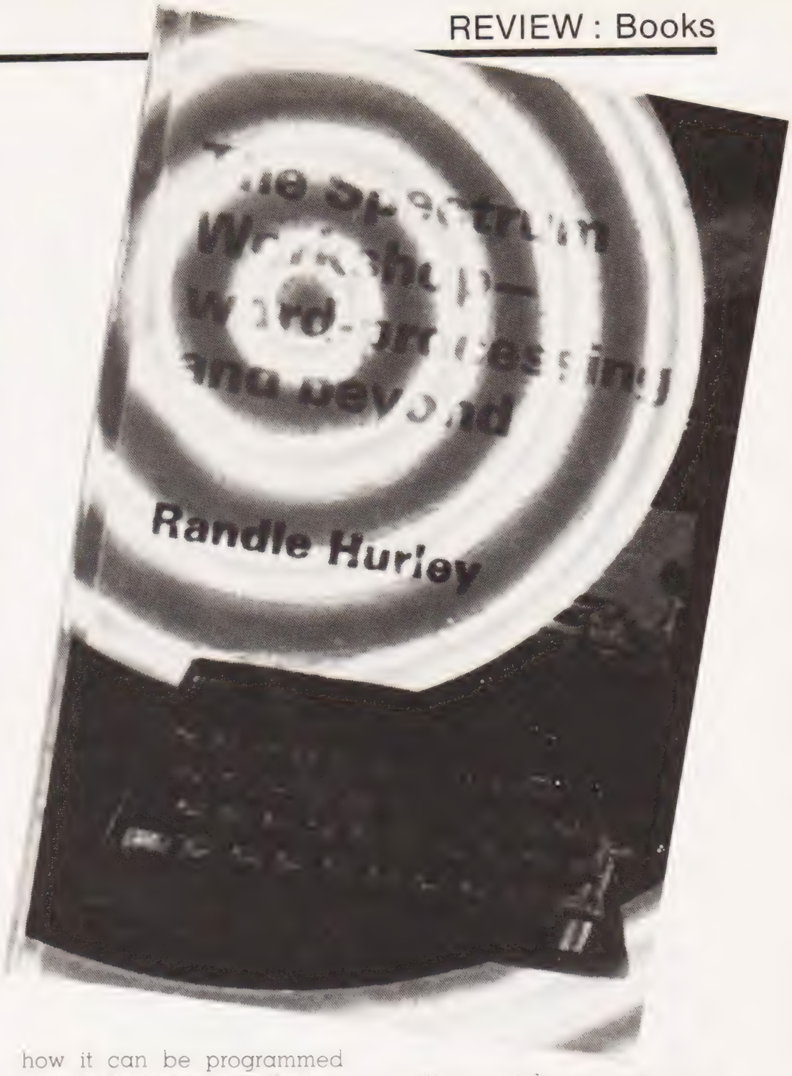
There is a need for books that give clear, careful and gradual introductions to BASIC programming. Since BASIC programmers themselves have not

been notably successful in producing them, it might be an idea for someone to involve the educational technologists in the process. They might also tackle another failing of the present book, which is that it is all about how to write programs, with nothing about why they might be written; all about technique with nothing about applications; all medium and no message.

I also tend to think that a computer-assisted learning (CAL) program is a better way to introduce programming than is a book. There are certain aspects of programming that are dynamic and which can be illustrated and explained better by a mobile simulation on a video screen than by a book. And having acquired a micro, why not use it as an aid to learning? Anyway, there is a long tradition of CAL programs for BASIC on micros going back to the first PETs all those years ago. Of course, piracy of cassette software has affected its development, but that is another matter. I doubt if many people will use the photocopier to pirate this book.

I admit that I am not the best person to review this book, but nor is anyone with some grasp of BASIC. The ideal reviewer would be an intelligent 10- or 12-year old just getting hooked on computers, although I still think I know what the verdict would be.

The Spectrum Workshop — Word Processing and Beyond by Randle Hurley is a natural to examine after the previous pair. In it, programs are developed for two word processors, a filing system and for merging information from the filing system into documents produced with one of the word processing programs. The programs are developed using segments and procedures that provide a kit of parts, and the way that each program is developed is explained carefully at each stage. In this way you can understand and develop the author's programs or your own variations on them, and can learn a good deal about BASIC and writing programs at the same time. This meets the twin objectives of providing a powerful set of programs for the Spectrum and demonstrating how it can be programmed near to its capacity. So here is an end product and motivation for learning.



The author uses BASIC to develop his first word processing program. Exploring its capabilities and shortcomings then leads him to write the second one in machine code. The approach to the assembly code programming is original, and its basis is to start with BASIC, but only with those instructions that correspond directly to one, or to a very few, assembly code instructions. The bridge from BASIC to assembly code then becomes short and is easily crossed.

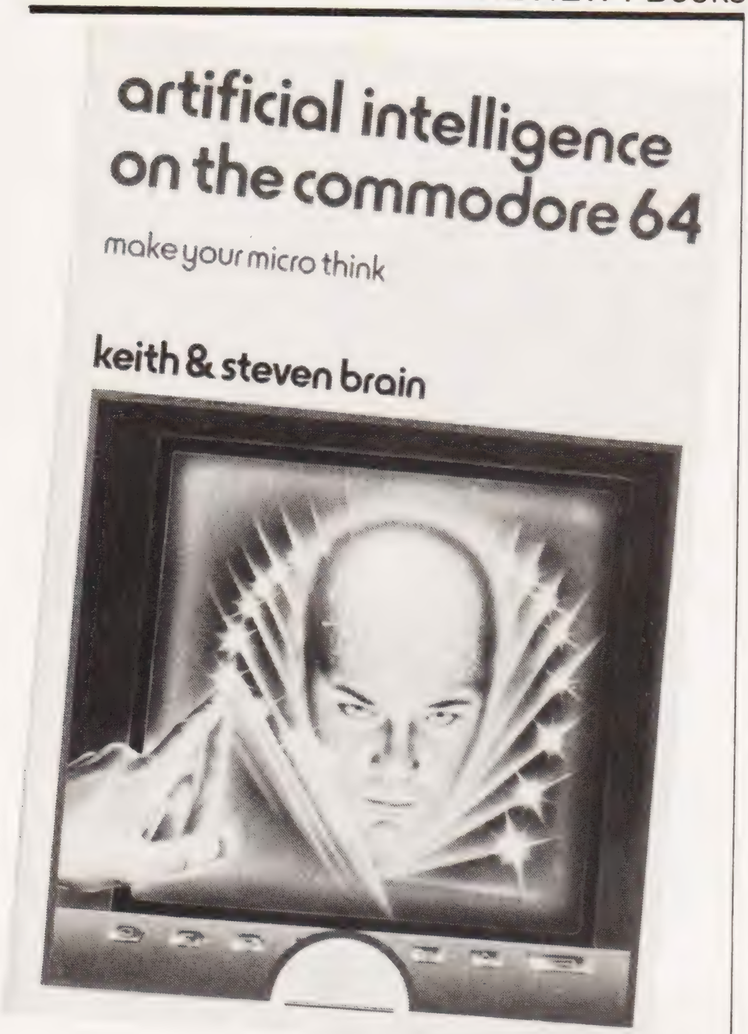
I am very impressed by the book. Incidentally, if you just want the programs for your Spectrum, they are available from the publisher.

Artificial Intelligence On The Commodore 64 by Keith and Steven Brain has the subtitle 'Make Your Micro Think'. Both title and subtitle overstate their cases, but this doesn't prevent the book from being rather good when examined by its own standards. It is neatly organised and has been well thought out.

The contents of a chapter can appear innocuous. The programs in the book have all been seen before. There is a pattern-matching program, a learning program as on the BBC's Welcome tape, and a program for holding conversations of the kind that has been around for as long as micros have, and even longer than that. But each program leads on to, and contributes to, its successor. The culmination is an impressive program built on all the others, that is somehow greater than the sum of its constituents. The way in which it has been built is also instructive.

The first step is made with what is essentially the familiar 'artist's drawing' program, which drives a 'drawing head' in various directions on the screen. Nothing new here, but do not skip anything because routines for later use are being given. Next, ways of giving instructions to the 'drawing head' using words are investigated, with the command words recognised by pattern-matching methocs. This leads on to recognising simple sentences, and then to a program that not only recognises sentences, but constructs answers or responses to them. This is done by making substitutions in the input sentence to provide the response.

Then a program that can



make decisions about objects presented to it by asking a series of questions and acting on the replies is presented. This leads from simple structures for the questioning to the tree structures that underly so many aspects of artificial intelligence. Ways of making the object recognition system acquire further information so as to be able to recognise objects outside its existing repertoire are examined next. Finally, all of this is put together in an ELIZA-style conversational program.

I applaud the structure and development of the book. My quibbles are minor. The first is that the programs do not take advantage of the special features of the Commodore 64. The second is that the authors treat the terminology of artificial intelligence with scant respect. Their versions of expert systems, natural language recognition and artificial intelligence barely reach the foothills of what are recognised as the formal subjects with these names. To illustrate this for only one topic, Professor Weizebaum has been at great pains to point out that his ELIZA is not really intelligent, and should not be taken as an example of artificial intelligence. Yet here it is as the jewel in the crown of this book which claims to be on this subject.

The book won't make the Commodore 64 think, but it will make us think if we study it well. Incidentally, a forthcoming book by Mike James deals much more penetratingly with artificial intelligence for micros.

This month's books are:

Word Processing for Beginners by Susan Curran (Granada), £5.95, 172 pages.

Writing Your First Programs by John Parry (Piccadilly Press), £4.95, 71 pages.

The Spectrum Workshop by Randle Hurley (McGraw-Hill), £6.95, 142 pages.

Artificial Intelligence On The Commodore 64 by Keith and Steven Brain (Sunshine), £6.95, 144 pages.

LEARNING FORTH PART 7

Paul Gardner

Or, time to own up . . .

In the writing and production of any series of articles it is almost inevitable that a few mistakes will creep in. So that you can have a completely error free collection, we shall put right all the errors that we have found.

The series began well without any mistakes in Parts 1 or 2 and only a tiny mistake in Part 3, on page 69 of the January 1984 issue. In the ninth line of the second column, the word **compiled** should read completed.

Unfortunately, Part 4, February 1984 is a different story. It appears that the blame lies in the use of a 'temporary glue' that gave the impression that inserts were firmly stuck on the page, only to come unstuck somewhere between here and the printers! (That's our story anyway . . .) So here follows, with our sincere apologies, a list of the corrections that should make sense of Part 4.

On page 37, second column, line 41 read:

TAKE IN CRA

when it should have been the FORTH expression:

20 ARRAY COUNTERS

The bottom of the first column on page 38 saw yet another mysterious phrase:

CRA 15

when it should have read:

6 TEST

Finally, the bottom of the first column on page 40:

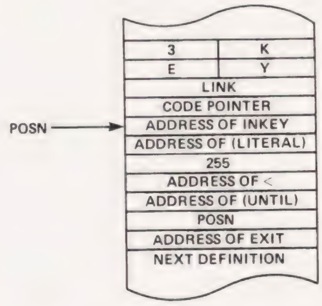
0 RAND 32

should simply have read:

0 RAND

Part 5, March 1984, saw only one small mistake. In the penultimate paragraph of the first column, page 50, there were two references to the **DICTIONARY BUFFER**. These should have referred to the **WORD BUFFER**.

Part 6 of the series, April 1984, saw three mistakes. The first was a misplaced diagram as Fig. 1 on page 32. The correct diagram was printed in the following issue but for those of you who missed it, here it is again:



On page 34 of that issue, in the first column, I omitted the definition of a FORTH word that should have preceded **(JUMPBACK).** I should have included the following:

TOHERE HERE IMMEDIATE

And finally, in the definition of **TEST** in the second column of page 34, the word **ENDRANGE** should have been **ENDIN.** (Next month we'll end the series by correcting all the mistakes found in Part 7!)

CP/M SOFTWARE DIRECTORY

ACCOUNTS

ACTPAK
Solicitor Systems and Services,
Systems House,
127 High Street,
Beckenham,
Kent BR3 1AG
Phone 01-658 3937 and 3938
Contact: Mr. W. Freeman
Price: £1500 (includes three days on-site training).

AUTOMATIC INVOICING
Bromley Computer Consultancy,
244a High Street,
Bromley,
Kent BR1 1PQ
Phone: 01-697 8933
Telex: 896691
Contact: Mr. A. Berridge
Price: £1000

BETPLAN
Bromley Computer Consultancy,
244a High Street,
Bromley,
Kent BR1 1PQ
Phone: 01-697 8933
Telex: 896691
Contact: Mr. A. Berridge
Price: £2000

BILLFLOW
Great Northern Computer Services Ltd,
16 Town Street,
Horsforth,
Leeds LS18 4RJ
Phone: 0532 589980
Contact: Sally Bagnal or Paul Rayner
Price: £395

BOSS-PAYPLAN
Bromley Computer Consultancy,
244a High Street,
Bromley,
Kent BR1 1PQ
Phone: 01-697 8933
Telex: 896691
Contact: Mr. A. Berridge
Price: £1000

BOSS-PAYROLL
Bromley Computer Consultancy,
244a High Street,
Bromley,
Kent BR1 1PQ
Phone: 01-697 8933
Telex: 896691
Contact: Mr. A. Berridge
Price: £500

BUSINESS DESK
Paxton Computers Ltd,
28 New Street,
St. Neots,
Huntingdon,
Cambridgeshire PE19 1AJ
Phone: 0480 213785
Telex: 32720
Contact: Jayson Brown
Price: From £595 to £2000

COSTING LEDGERS– BUSINESS MANAGEMENT SYSTEMS
Peachtree Software International,
43-53 Moorbridge Road,
Maidenhead,
Berkshire
Phone: 0628 32711
Contact: Sales Manager
Price: £600

EXACT
Bickerton Computer Centre,
38a Abbey Foregate,
Shrewsbury,
Shropshire SY2 6BL
Phone: 0743 68167
Contact: Richard Bickerton
Price: £750

GENERAL LEDGER SYSTEM
Interface Micro Systems,
160 High Street,
Southend-on-Sea,
Essex SS1 1JX
Phone: 0702 69961
Contact: Terry Willingham
Price: £350

INTERACC
Software Aids International Ltd,
16 Norval Road,
North Wembley,
Middlesex HA0 3TE
Phone: 01-904 8139
Contact: Mr. D.V. Bull
Price: From £200

INVENTORY MANAGEMENT – BUSINESS MANAGEMENT SYSTEMS
Peachtree Software International,
43-53 Moorbridge Road,
Maidenhead,
Berkshire
Phone: 0628 32711
Contact: Sales Manager
Price: £600

ISL
The Byte Shop (Birmingham) Ltd
94-96 Hurst Street,
Birmingham B5 4TD
Phone: 021-622 3165
Contact: Mr. Southern
Price: £1100

NOMINAL AND PURCHASE LEDGER
Bickerton Computer Centre,
38a Abbey Foregate,
Shrewsbury,
Shropshire SY2 6BL
Phone: 0743 68167
Contact: Richard Bickerton
Price: £300

NOMINAL LEDGER
Compact Accounting Services Ltd
Cape House,
Cape PLace,
Dorking,
Surrey
Phone: 0306 887373
Contact: David Parson
Price: £325

NOMINAL LEDGER – BUSINESS MANAGEMENT SYSTEMS
Peachtree Software International,
43-53 Moorbridge Road,
Maidenhead,
Berkshire
Phone: 0628 32711
Contact: Sales Manager
Price: £600

PAYROLL
Bickerton Computer Centre,
38a Abbey Foregate,
Shrewsbury,
Shropshire SY2 6BL
Phone: 0743 68167
Contact: Richard Bickerton
Price: £200

PAYROLL
Compact Accounting Services Ltd
Cape House,
Cape PLace,
Dorking,
Surrey
Phone: 0306 887373
Contact: David Parson
Price: £425

PAYROLL
Dataflow (UK) Ltd,
Unit 18,
Central Trading Estate,
Staines,
Middlesex TW18 4UX
Phone: 0784 54171
Contact: Jane Rising
Price: £500

PAYROLL – BUSINESS MANAGEMENT SYSTEMS
Peachtree Software International,
43-53 Moorbridge Road,
Maidenhead,
Berkshire
Phone: 0628 32711
Contact: Sales Manager
Price: £600

PURCHASE LEDGER
Bromley Computer Consultancy,
244a High Street,
Bromley,
Kent BR1 1PQ
Phone: 01-697 8933
Telex: 896691
Contact: Mr. A. Berridge
Price: £500

PURCHASE LEDGER
Compact Accounting Services Ltd
Cape House,
Cape PLace,
Dorking,
Surrey
Phone: 0306 887373
Contact: David Parson
Price: £325

PURCHASE LEDGER
Dataflow (UK) Ltd,
Unit 18,
Central Trading Estate,
Staines,
Middlesex TW18 4UX
Phone: 0784 54171
Contact: Jane Rising
Price: £500

PURCHASE LEDGER – BUSINESS MANAGEMENT SYSYEMS
Peachtree Software International,
43-53 Moorbridge Road,
Maidenhead,
Berkshire
Phone: 0628 32711
Contact: Sales Manager
Price: £600

PURCHASE LEDGER SYSTEM
Interface Micro Systems,
160 High Street,
Southend-on-Sea,
Essex SS1 1JX
Phone: 0702 69961
Contact: Terry Willingham
Price: £350

SALES LEDGER – BUSINESS MANAGEMENT SYSTEMS
Peachtree Software International,
43-53 Moorbridge Road,
Maidenhead,
Berkshire
Phone: 0628 32711
Contact: Sales Manager
Price: £600

SALES LEDGER SYSTEM
Interface Micro Systems,
160 High Street,
Southend-on-Sea,
Essex SS1 1JX
Phone: 0702 69961
Contact: Terry Willingham
Price: £350

DATABASES

BT-80
This is a Digital Research product and should be available from most software suppliers.

CARDBOX
Caxton Software Publishing Company Ltd,
10-14 Bedford Street,
Covent Garden,
London WC2E 9HE
Phone: 01-379 6502
Contact: W.D. Barrow
Price: £195

CONDOR
MOM Systems Ltd,
40-41 Windmill Street,
Gravesend,
Kent DA12 1BA
Phone: 0474 57746
Telex: 965492
Price: From £195 to £650

DATAFLEX
Equinox Computers,
16 Anning Street,
New Inn Yard,
London EC2A 3HB
Phone: 01-739 2387
Telex: 27341
Contact: Mike Kusmirak
Price: Single-user £495, multi-user £595

DATAFLOW
Dataflow (UK) Ltd,
Unit 18,
Central Trading Estate,
Staines,
Middlesex TW18 4UX
Phone: 0784 54171
Contact: Jane Rising

DATAFLOW
Great Northern Computer Services Ltd,
16 Town Street,
Horsforth,
Leeds LS18 4RJ
Phone: 0532 589980
Contact: Sally Bagnal or Paul Rayner
Price: £185

DATA MANAGEMENT SYSTEM
Compsoft Ltd,
Hallams Court,
Shamley Green,
Near Guidford,
Near Guildford,
Surrey
Phone: 0483 898545
Contact: Heather Kearsley,
Price: £195

MICROPEN
Transam Microsystems Ltd,
59-61 Theobalds Road,
London WC1
Phone: 01-404 4554
Price: £125

DATASTAR
This is a MicroPro International product and should be available from most software suppliers.

DATA STORE ONE SYSTEM
Interface Micro Systems,
160 High Street,
Southend-on-Sea,
Essex SS1 1JX
Phone: 0702 69961
Contact: Terry Willingham
Price: £100

dBASE II
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

dBASE II DATABASE MANAGEMENT SYSTEM
Sussex Microsystems Ltd,
43 East Street,
Horsham,
West Sussex
Phone: 0403 68071
Contact: Keith Gale
Price: "under £500"

FILEFORCE
Transam Microsystems Ltd,
59-61 Theobalds Road,
London WC1
Phone: 01-404 4554
Price: £250

GUARDIAN
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

MAILING ADDRESS
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

MAILING SYSTEM
Interface Micro Systems,
160 High Street,
Southend-on-Sea,
Essex SS1 1JX
Phone: 0702 69961
Contact: Terry Willingham
Price: £100

MAILMASS
Vision Associates Ltd,
57 Woodham Lane,
New Haw,
Weybridge,
Surrey,
Phone: 0932 55932
Contact: Graeme Sleeman
Price:

PRISM
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

TRENDISK/1c
Microtrend UK,
PO Box 51,
Pateley Bridge,
Harrogate,
North Yorkshire HG3 5DF
Phone: 0423 711878
Contact: Ian Cummings
Price: £150

GRAPHICS

DISPLAY MANAGER
This is a Digital Research product and should be available from most software suppliers.

DATAPLOT III
Grafox Ltd,
35 St. Clements,
Oxford OX4 1AB
Phone: 0865 242597
Telex: 83147
Contact: Geordie Herbert
Price: £495

DATAPLOT+
Grafox Ltd,
35 St. Clements,
Oxford OX4 1AB
Phone: 0865 242597
Telex: 83147
Contact: Geordie Herbert
Price: £195

LANGUAGES

ALGOL-60
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

APL V80
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

BAZIC
Xitan Systems Ltd,
23 Cumberland Place,
Southampton,
Hampshire SO1 2BB
Phone: 0703 871211
Telex: 47388
Contact: Jane Dards
Price: £124

C-BASIC
This is a Microsoft product and should be available from most software suppliers.

CBASIC COMPILER
Xitan Systems Ltd,
23 Cumberland Place,
Southampton,
Hampshire SO1 2BB
Phone: 0703 871211
Telex: 47388
Contact: Jane Dards
Price: £333

CIS COBOL
Xitan Systems Ltd,
23 Cumberland Place,
Southampton,
Hampshire SO1 2BB
Phone: 0703 871211
Telex: 47388
Contact: Jane Dards
Price: £425

COBOL-80
This is a Microsoft product and should be available from most software suppliers.

COMAL
Transam Microsystems Ltd,
59-61 Theobalds Road,
London WC1
Phone: 01-404 4554
Price: £145

POLYFORTH
Computer Solutions Ltd,
Treway House,
Hanworth Lane,
Chertsey,
Surrey KT16 9LA
Phone: 093 28 65292
Contact: Chris Stephens
Price: From £750 to £2750

FORTRAN-80
This is a Microsoft product and should be available from most software suppliers.

JRT PASCAL
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

KBASIC
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

LISP-80
Transam Microsystems Ltd,
59-61 Theobalds Road,
London WC1
Phone: 01-404 4554
Price: £50

M-BASIC
This is a Microsoft product and should be available from most software suppliers.

PASCAL/M
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

PASCAL/MT
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

PASCAL/MT+
Xitan Systems Ltd,
23 Cumberland Place,
Southampton,
Hampshire SO1 2BB
Phone: 0703 871211
Telex: 47388
Contact: Jane Dards
Price: £233

PASCAL/Z
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

PRO PASCAL
Prospero Software,
37 Gwendolen Avenue,
London SW15 6EP
Phone: 01-785 6848
Contact: Tony Hetherington
Price: £220

PRO FORTRAN
Prospero Software,
37 Gwendolen Avenue,
London SW15 6EP
Phone: 01-785 6848
Contact: Tony Hetherington
Price: £220

UCSD PASCAL
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

MODELLING

CALCSTAR
This is a MicroPro International product and should be available from most software suppliers.

FASTPLAN
Transam Microsystems Ltd,
59-61 Theobalds Road,
London WC1
Phone: 01-404 4554
Price: £395

PEACHCALC
Peachtree Software International,
43-53 Moorbridge Road,
Maidenhead,
Berkshire
Phone: 0628 32711
Contact: Sales Manager
Price: £200

PLAN80
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

SUPERCALC
Tamsys Ltd,
12a Sheet Street,
Windsor,
Berkshire SL4 1BG
Phone: 075 35 56747
Contact: David Atkinson
Price: £129

SUPERCALC I
Xitan Systems Ltd,
23 Cumberland Place,
Southampton,
Hampshire SO1 2BB
Phone: 0703 871211
Telex: 47388
Contact: Jane Dards
Price: £126

SUPERCALC II
Xitan Systems Ltd,
23 Cumberland Place,
Southampton,
Hampshire SO1 2BB
Phone: 0703 871211
Telex: 47388
Contact: Jane Dards
Price: £195

PROGRAMMING

ACCESS MANAGER
This is a Digital Research product and should be available from most software suppliers.

ANIMATOR
Xitan Systems Ltd,
23 Cumberland Place,
Southampton,
Hampshire SO1 2BB
Phone: 0703 871211
Telex: 47388
Contact: Jane Dards
Price: From £225

BASIC UTILITY DISK
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

BASKAM
Great Northern Computer Services Ltd,
16 Town Street,
Horsforth,
Leeds LS18 4RJ
Phone: 0532 589980
Contact: Sally Bagnal or Paul Rayner
Price: £95

COBOL DEVELOPMENT SYSTEM
Interface Micro Systems,
160 High Street,
Southend-on-Sea,
Essex SS1 1JX
Phone: 0702 69961
Contact: Terry Willingham
Price: Full system £550

CLIP
Tamsys Ltd,
12a Sheet Street,
Windsor,
Berkshire SL4 1BG
Phone: 075 35 56747
Contact: David Atkinson
Price: £95

FORMS UTILITY GENERATOR
Interface Micro Systems,
160 High Street,
Southend-on-Sea,
Essex SS1 1JX
Phone: 0702 69961
Contact: Terry Willingham
Price: £120

PSORT
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

QSORT
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

SPP
Xitan Systems Ltd,
23 Cumberland Place,
Southampton,
Hampshire SO1 2BB
Phone: 0703 871211
Telex: 47388
Contact: Jane Dards
Price: £133

T/MAKER II
Transam Microsystems Ltd,
59-61 Theobalds Road,
London WC1
Phone: 01-404 4554
Price: £165

TRANSAM SCIENTIFIC SUBROUTINES
Transam Microsystems Ltd,
59-61 Theobalds Road,
London WC1
Phone: 01-404 4554
Price: £495

TYPESTAR
Sussex Microsystems Ltd,
43 East Street,
Horsham,
West Sussex
Phone: 0403 68071
Contact: Keith Gale
Price: £95

XLT86
Xitan Systems Ltd,
23 Cumberland Place,
Southampton,
Hampshire SO1 2BB
Phone: 0703 871211
Telex: 47388
Contact: Jane Dards
Price: £100

WORD PROCESSING

BENCHMARK
Interface Micro Systems,
160 High Street,
Southend-on-Sea,
Essex SS1 1JX
Phone: 0702 69961
Contact: Terry Willingham

BRITISH SPELLGUARD
Vision Associates Ltd,
57 Woodham Lane,
New Haw,
Weybridge,
Surrey,
Phone: 0932 55932
Contact: Graeme Sleeman
Price: £179

LETTERIGHT
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

LEXICOM/2
Microtrend UK,
PO Box 51,
Pateley Bridge,
Harrogate,
North Yorkshire HG3 5DF
Phone: 0423 711878
Contact: Ian Cummings
Price: £350

MAGIC WAND
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

MAILMERGE
This is a MicroPro International product and should be available from most software suppliers.

PEACHTEXT
Peachtree Software International,
43-53 Moorbridge Road,
Maidenhead,
Berkshire
Phone: 0628 32711
Contact: Sales Manager
Price: £250

SPELL
Transam Microsystems Ltd,
59-61 Theobalds Road,
London WC1
Phone: 01-404 4554
Price: £55

SPELLBINDER
Transam Microsystems Ltd,
59-61 Theobalds Road,
London WC1
Phone: 01-404 4554
Price: £275

SPELLING PROOFREADER
Peachtree Software International,
43-53 Moorbridge Road,
Maidenhead,
Berkshire
Phone: 0628 32711
Contact: Sales Manager
Price: £600

SPELLSTAR
This is a MicroPro International product and should be available from most software suppliers.

TEXTWRITER III
Lifeboat Associates Ltd,
PO Box 125,
London WC2H 9LU
Phone: 01-836 9028
Telex: 893709

WORD PROCESSING SYSTEM
Interface Micro Systems,
160 High Street,
Southend-on-Sea,
Essex SS1 1JX
Phone: 0702 69961
Contact: Terry Willingham
Price: £100

WORDSTAR
This is a MicroPro International product and should be available from most software suppliers.

NEXT MONTH

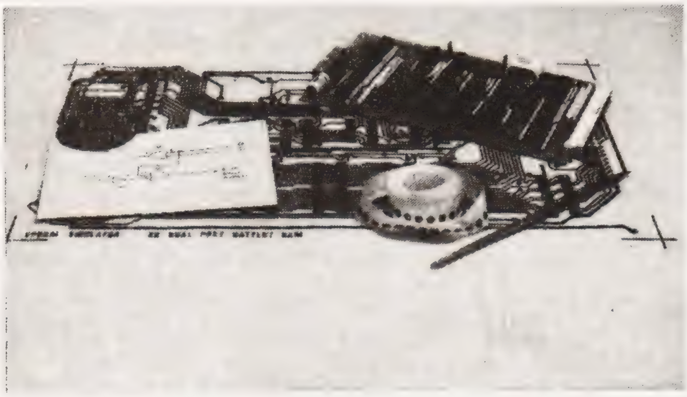
AN ARGUS SPECIALIST PUBLICATION

electronics today INTERNATIONAL

Accept an Imitation!

Next month we shall be asking all our readers to accept an imitation — not for ETI, because we know that would never satisfy you — but for the common-or-garden EPROM. The project concerned is an EPROM emulator, and it is designed to work using a very large variety of host computers.

The idea of an EPROM emulator is that you should plug it into a system (which could be any sort of digital circuit, not just a microcomputer) in the place of an EPROM; to the circuit, the system 'looks like' an EPROM; however, the emulator is linked to a host computer, and this can be used to modify the emulator contents, with the minimum of fuss. Once you've got the data you want in the emulator, all you do next is to program this into your EPROM using the ETI EPROM programmer — couldn't be simpler, could it?



Communications Satellites

The first communications satellite — Telstar — was successfully launched over 20 years ago. There was, at the time, a popular 'hit', record by the Tornadoes, named after Telstar! Nowadays, satellite communications seem more mundane — do many people realise that, sometimes, when they dial a foreign number, their call will be going through space? Next month we'll be taking a look at the technology involved in satellite communications.

MOSFET Power Amplifier

As you will see on page 24, John Linsley Hood has commenced his description of the 'Audio Design' amplifier system with the preamp section. Next month it will be the turn of the power amplifier . . . can the Editor's neighbours possibly wait until then?

Security

The July issue of ETI has been designated a 'Security Special', and this means that there will be features and projects aimed at helping you make your house or flat — and car — more protected from unwelcome attention. We hope to arrange a couple of rather good readers' offers, but details of these are still under negotiation as we go to press.

ALL IN THE JULY ISSUE OF ETI — ON SALE FRIDAY JUNE 1st. PLACE YOUR ORDER NOW OR RISK MISSING OUT!

A FAST SORT

Don Thomasson

When we discussed the pulling apart of machine code a couple of issues ago, we pointed out that a good sort routine would be very useful. Here's a suitable candidate, with other applications a possibility.

When data has to be sorted, there is a tendency to think first of the bubble sort, but that is rather slow. The method used in the program descibed here is much faster, and may approach the theoretical limit of sorting speed. The program is specialised for the production of sorted lists of address pairs, a function that is almost essential when a large machine code program is being analysed (as discussed in the article 'Disassembly Techniques' in CT April 84), but the individual routines can be applied on a much wider basis.

THE OVERALL PROGRAM

Listing 1 covers a complete set of routines for setting up and modifying of data, displaying the lists on the screen and printing them out. Lines 100-240 form the backbone of the program, which dimensions arrays, presents a menu, and jumps to the required routine.

The program is written for the BBC Model B, and the array size may need adjusting to the limitations of a particular machine or to special user requirements. All three arrays should be of the same size.

DATA ENTRY

Data entry is handled by lines 260-320, using a subroutine at 560-620. There are two entry points from the menu. Entry at line 260 resets N, the count of entries, and clears array A. this entry is used when starting a new list. Entry at line 280 leaves the current list intact, allowing further entries to be added.

If the subroutine is ignored, the data entry process is simple enough. N is incremented before being used as a pointer to array A, since it will later be used to terminate action when all the array elements have been handled. That is why N is initialised to −1, rather than to zero. With machines that will not accept a zero subscript, such as the Spectrum, some careful adjustments to statements involving N will be necessary.

If the subroutine returns with "G" as the only input character, the routine drops out to the menu.

The subroutine is more complex, being specialised to the particular task for which the program was written. The input consists of pairs of hexadecimal addresses. The first address may be a jump destination, the second the point where the jump is located. The entries are to be sorted in the order of the first addresses, the second addresses being taken in order where the first addresses are equal.

The method adopted here combines the two addresses into a single number, which can be as great as $4.3 * 10^9$. This demands a floating point system with a 32-bit mantissa, and for machines which lack that facility it may be easier to treat the entries as strings, though this will usually be slower.

If an entry does not contain nine characters, VDU 7 gives a 'beep' to warn you that you have made an error, and the input is ignored. If the entry consists only of the letter 'G', the subroutine returns. Otherwise, the EVAL function is used to convert the two addresses to numeric form, and the two numbers are then combined.

Many alternative forms of input subroutine can be used. This one illustrates a way of making the most of available storage space.

DISPLAY & PRINT

A common routine is used for screen display and print, but whereas the screen can only accommodate three columns of data entries with convenience, the printer can easily manage seven. Entry at line 340 switches the printer on and sets E to 7, calls the common routine and then switches the printer off. Entry at line 400 sets E to 3, and the common routine follows. It need scarcely be said that E determines the number of columns . . .

The common routine begins by calculating K, which is the number of lines in the list. If there are 50 entries and three columns, N=49, and INT(N/E)+1=17. There will be two columns of 17 entries and a third column of 15 entries. Two nested loops put this into practice.

The outer loop (index P), deals with the lines of the output. In the case stated above, it deals with lines 0 to 16. The inner loop (index Q) deals with the columns within a line, adding K to the pointer P +Q*K as one column succeeds another. The result is that the entries in a given column are in sequence, reading downwards, and the next column continues the sequence, and so on. This is a routine that may well be of use in other circumstances.

SORTING

The actual sorting routine occupies lines 640 to 880. The method involves first splitting the contents of array A between arrays B and C. The first item is put into array B, and if the next item is greater in magnitude it is again put into array B. Action switches between B and C whenever the next item is smaller than the last, ie when ascending sequence is broken. For example, if A holds:

```
4 A 5 E 7 6 2 9 1 D 3 8 B C F
```

then B and C hold:

```
4 A 7 2 9 3 8 B C F
5 E 6 1 D
```

```
100 N=-1
110 CLS
120 DIM A(1000)
130 DIM B(1000),C(1000)
140 CLS:PRINT "Select Mode"
150 PRINT TAB(5);"1; Enter New List"
160 PRINT TAB(5);"2; Continue Old List"
170 PRINT TAB(5);"3; Delete Entry"
180 PRINT TAB(5);"4; Sort"
190 PRINT TAB(5);"5; Output"
200 PRINT TAB(5);"6; Print"
210 INPUT "INPUT 1 - 6",F
220 IF F<1 OR F>6 THEN 140
230 ON F GOSUB 260,280,890,640,400,340
240 GOTO 140
250 REM Data entry routine.
260 N=-1
270 FOR X=0 TO 1000:A(X)=0
280 GOSUB 560
290 IF LEFT$(D$,1)="G" RETURN
300 N=N+1
310 A(N)=H
320 GOTO 280
330 REM Print routine.
340 VDU 2
350 E = 7
360 GOSUB 420
370 VDU3
380 RETURN
390 REM Display routine.
400 CLS
410 E = 3
420 K=INT(N/E)+1
430 FOR P=0 TO (K-1)
440 FOR Q=0 TO E-1
450 R=A(P+Q*K)
460 IF (P+Q*K)>N THEN 500
470 H=INT(R/65536)
480 J=R-65536*H
490 PRINT TAB(10*Q);~H;" ";~J;
500 NEXT Q
510 PRINT
520 NEXT P
530 PRINT
540 INPUT X:RETURN
550 REM Input routine.
560 INPUT D$
570 IF LEFT$(D$,1)="G" RETURN
580 IF LEN(D$)<>9 THEN VDU7:GOTO 560
590 J=EVAL("&"+LEFT$(D$,4))
600 H=EVAL("&"+RIGHT$(D$,4))
610 H=H+65536*J
620 RETURN
630 REM Sort routine
640 FB=0:PRINT"Sorting"
650 NA=0:NB=0:NC=0:FA=0
660 FOR X=0 TO 1000:B(X)=0:C(X)=0:NEXTX
670 HA=A(NA)
680 IF FA=0 THEN 710
690 C(NC)=HA:NC=NC+1
700 GOTO 720
710 B(NB)=HA:NB=NB+1
720 NA=NA+1:IF NA>N THEN 760
730 IF HA<=A(NA) THEN 670
740 FA=(FA+1)AND 1
750 GOTO 670
760 NA=0:NB=0:NC=0:HA=0
770 IF B(NB)=0 THEN 840
780 IF C(NC)=0 THEN 830
790 IF B(NB)>=HA AND C(NC)>=HA THEN 820
800 IF B(NB)>=HA THEN 830
810 IF C(NC)>=HA THEN 840
820 IF B(NB)>C(NC) THEN 840
830 HA=B(NB):NB=NB+1:GOTO850
840 HA=C(NC):NC=NC+1
850 A(NA) = HA:NA=NA+1
860 IF NA <= N THEN 770
870 FB=FB+1:PRINT FB:IF C(0)=0 THEN RETURN
880 GOTO 650
890 GOSUB560
900 C=-1
910 FOR X=0 TO N
920 IF H=A(X) THEN C=X
930 NEXT X
940 IF C<0 THEN PRINT "Not found":P$=INKEY$(150)
:RETURN
950 FOR X=C TO (N-1)
960 A(X)=A(X+1)
970 NEXT
980 N=N-1
990 RETURN
```

The contents of B and C are then merged back into A, using the following rules:

- If the next item in B and the next item in C are both greater than the last item copied into A, the smaller of the B and C items is taken.
- If only the B item is greater than the last item copied into A, the B item is taken.
- If only the C item is greater than the last item copied into A, the C item is taken.
- If neither the next B or C item is greater than the last item copied into A, the smaller item is taken. (This starts a new sequence.)

Applying these rules to the data quoted above, A will hold:

4 5 A E 6 7 1 2 9 D 3 8 B C F

There are at least two correctly ordered entries in each sequence after this first pass, and the longest ordered sequence contains five items. Splitting again into B and C:

4 5 A E 1 2 9 D
6 7 3 8 B C F

Repeating the merge cycle gives:

4 5 6 7 A E 1 2 3 8 9 B C D F

Ordered sequences at this stage must be at least four items long, and the longest ordered sequence actually contains nine items. The third split gives:

4 5 6 7 A E
1 2 3 8 9 B C D F

The merge completes the sort:

1 2 3 4 5 6 7 8 9 A B C D E F

The subsequent split puts all the data in B, and C is empty, and we can use that fact to indicate that the sort is complete.

To give reassurance while the sort is in progress, the iteration number is displayed on the screen, but this is not essential.

As to speed, 878 items were sorted in 4½ minutes, taking nine iterations. This is appreciably faster than a bubble sort. An even better performance is shown when some extra items have been added and the sort is repeated, no more than a minute being taken in some cases.

ERROR CORRECTION

Despite the safety check on inputs, it was found that some errors did creep in when setting up lists of up to 900 items, so a correction routine was added in lines 890 to 990. This uses the input subroutine to define the item to be removed, and array A is then scanned in search of a match to this data. A match is marked by setting C to the subscript for the matching entry. If C remains −1, no match has been found, and this is reported briefly but if C is not −1 a copy action moves all items below the matching item up one place. The matching item is thus erased, and with N decremented the remaining list is intact. A corrected entry may then be added to replace the erroneous one, and a quick sort will put it in its place.

CONCLUSION

Though of great value in its specialised role, the program can readily be adapted to other purposes. Since it was to run on the BBC machine, it is natural that the capabilities of that machine were put to good use, but for those who wish to convert for other machines the following interpretations may be useful:

VDU 2: Printer on
VDU 3: Printer off
Tilde: Print in hex (see line 490)
INKEY$(X): Wait X/100 seconds.

LORDS OF TIME

Joins our range of acclaimed pure-text puzzle adventures, at £9.90, for:

BBC 32K ***COMMODORE 64*** ***SPECTRUM*** 48K ***LYNX*** 48K ***NASCOM*** 32K ***ORIC*** 48K ***ATARI*** 32K

ADVENTURE REVIEWS

"Adventures which have a fast response time, are spectacular in the amount of detail and number of locations, and are available to cassette owners . . Simply smashing!"
– Soft, Sept 83

"**Colossal Adventure** is included in Practical Computing's top ten games choice for 1983: "Poetic, moving and tough as hell."
– PC, Dec 83

"**Colossal Adventure** . . For once here's a program that lives up to its name . . a masterful feat. Thoroughly recommended"
– Computer Choice, Dec 83

"**Colossal Adventure** is one of the best in its class. I would recommend it to any adventurer."
– Acorn User, Feb 84

"**Adventure Quest** . . This has always been one of the best adventures for me as it seems to contain the lot. In all it took me about eight months to solve."
– PCW, 18th Jan 84

"To sum up, **Adventure Quest** is a wonderful program, fast, exciting and challenging. If you like adventures then this one is for you"
– NILUG issue 1.3

"Colossal Adventure is simply superb . . For those who want to move onto another adventure of similar high quality, **Dungeon Adventure** is recommended. With more than 200 locations, 700 messages and 100 objects it will tease and delight!"
– Educational Computing, Nov 83

ADVENTURE REVIEWS

"Colossal Adventure . . undoubtedly the best Adventure game around. Level 9 Computing have worked wonders to cram all this into 32K . . Finally **Dungeon Adventure**, last but by no means least. This is the best of the lot – a truly massive adventure – you'll have to play it yourselves to belive it."
– CBM 64 Users Club Newsletter

"The puzzles are logical and the program is enthralling. **Snowball** is well worth the money which, for a computer program, is a high recommendation."
– Micro Adventurer, Dec 83

"**Snowball** . . As in all Level 9's adventures, the real pleasure comes not from scoring points but in exploring the world in which the game is set and learning about its denziens . . this program goes to prove that the mental pictures conjured up by a good textual adventure can be far more vivid than the graphics available on home computers."
– Which Micro?, Feb 84

"**Lords of Time.** This program, writen by newcomer Sue Gazzard, joins my favourite series and is an extremely good addition to Level 9's consistently good catalogue . . As we have come to expect from Level 9, the program is executed with wonderful style – none of those boring "You can't do that" messages! Highly recommended."
– PCW, 1st Feb 84

MIDDLE EARTH ADVENTURES

1: COLOSSAL ADVENTURE
A complete, full size version of the classic mainframe game "Adventure" with 70 bonus locations added

2: ADVENTURE QUEST
Centuries have passed since the time of Colossal Adventure and evil armies have invaded The Land. The way is long and dangerous: but with cunning you can overcome all obstacles on the way to the Black Tower, source of their demonic power, and destroy it

3: DUNGEON ADVENTURE
The trilogy is completed by this superb adventure, set in the Dungeons beneath the shattered Black Tower. A sense of humour is essential!

THE FIRST SILICON DREAM ADVENTURE

1: SNOWBALL
The first of Pete Austin's second trilogy. The giant colony starship, Snowball 9, has been sabotaged and is heading for the sun in this massive game with 7000 locations.

THE LORDS OF TIME SAGA

7: LORDS OF TIME
Our congratulations to Sue Gazzard for her super design for this new time travel adventure through the ages of world history. Chill to the Ice-age, go romin' with Caesar's legions, shed light on the Dark Ages, etc.

Price: £9.90 each (inclusive)

Level 9 adventures are available from good computer shops, or mail-order from us at no extra charge. Please send order, or SAE for catalogue, to

LEVEL 9 COMPUTING

Dept C, 229 Hughenden Road,
High Wycombe, Bucks HP13 5PG
Please describe your computer

DRAGON HINTS ON TINTS

Gordon Mills

Bored with the limited palette of the hi-res Dragon? We show you how to generate extra hues with our colour-by-numbers subroutine.

At first glance, the colour capabilities of the Dragon look unimpressive when compared to other low cost micros such as the VIC20 and the Spectrum. It is true that there are nine distinct colours available in the low resolution mode and 10 in the various hi-res modes: unfortunately, according to the manual, no more than four can be made available at one time on screen in hi-res modes and one of these has to be used as the background colour.

For those who are interested in hi-res graphics in colour, it soon becomes apparent that Mode 3 gives the greatest

1	BUFF
2	CYAN
3	MAGENTA
4	ORANGE
5	TURQUOISE
6	GAMBOGE
7	ULTRAMARINE
8	PINK
9	GREEN
10	PURPLE

Fig. 1 The colours by number.

flexibility. What is not so obvious is that Mode 3 with SCREEN 1,1 will allow the use of an additional six distinct colours. For convenience and for later reference, the complete set of colours is labelled and numbered as in Fig. 1. That these are the maximum available will become clear when I describe how to obtain them. The techniques for producing the extra colours do not work for SCREEN 1,0 nor for any of the other graphics modes.

TABLE 1

Colour number	Pattern of bits
2	0 1 0 1
3	1 0 1 0
4	1 1 1 1
5	0 1 1 0
6	0 1 1 1
7	1 0 0 1
8	1 0 1 1
9	1 1 0 1
10	1 1 1 0

HI RES ADDRESSING

The Mode 3 addressing system is based on the use of two bits to indicate the base colour to be found at each of two adjacent points. For example, if memory address 1536 (which stores information for the first eight points on the top line of the hi-res screen) contains the following eight bits:

00011011

the first two indicate BUFF (0 0) in the first two points, the second two indiate CYAN (0 1) in the third and fourth points, the third two indicate MAGENTA (1 0) in the fifth and sixth points and the final two indicate ORANGE (1 1) in the seventh and eighth points.

What makes it possible to produce additional colours is the interference between the latter three colours when placed immediately vertically adjacent. Instead of just considering a row of two, it becomes necessary to look at the pattern of bits two horizontally by two vertically. The other crucial factor which governs the colour produced by vertical mixing is which base colour is on an even numbered and which is on an odd numbered row. In the patterns indicated in Table 1, the top bits must be on an even numbered row and the lower bits must therefore be on an odd numbered line. Whilst the contents of Table 1 is of interest to the machine code specialist, the average user requires a simpler approach using standard BASIC routines.

COLOURED RECTANGLES

The LINE routine available in Dragon BASIC can only be used for drawing rectangles in the four basic colours. The subroutine in Listing 1 will allow rectangles to be drawn in any one of the 10 available colours. It works best on a BUFF background, on the other basic colour backgrounds there may be unwanted lines of different colour at the top of the rectangle. As with the LINE routine, the parameters required are X1, X2, Y1 and Y2 together with CL, the colour selected from the colour table.

Listing 2 is a small program to demonstrate the capabilities of the routine in Listing 1. To see how it looks on different backgrounds, simply add 6, 7 or 8 after PCLS in line 100. You can also try it for SCREEN 1,0 but you will be disappointed! Most of the combinations

```
100 PMODE 3.1: SCREEN 1.1: PCLS
110 Y1=48: Y2=144
120 FOR CL=1 TO 10
130 X1=CL*20: X2=X1+17
140 GOSUB 5000
150 NEXT CL
200 GOTO 200
```

Listing 2

in the latter mode give horizontal stripes. Incidentally, if you combine BUFF in SCREEN 1,1 with any of the other three colours, only horizontal stripes will be displayed.

ODDS AND . . .

While there are no limitations on the values of the start coordinates on screen of the rectangle you wish to draw, it will reduce overlap problems if X1 is even and X2 is odd. Y1 can be either odd or even and the only limitation is that there must be at least two horizontal lines in the rectangle to show one of the six extra colours. On some television sets, the gamboge and ultramarine may initially appear as dark grey and medium blue-grey respectively but suitable adjustments to the colour, contrast and brilliance controls may improve them to good rich colours. Personally, I find the turquoise, pink and green more attractive than any of the standard Dragon colours but this may be a matter of taste.

```
5000 REM COLOR SUB
5010 EL=CL: OL=CL: IF CL>4 THEN EL=INT((CL-1)/2): OL=CL-2*EL+2+(CL=7)+(EL=4)
5020 DD=OL-EL: K=1: IF INT(Y1/2)*2=Y1 THEN K=0
5030 FOR Q=Y1 TO. Y2: NC=EL-DD*K: K=(K=0): COLOR NC,1: LINE(X1,Q)-(X2,Q),PSET
5040 NEXT Q: RETURN
```

Listing 1

CLUB CALL

Fiona Eldridge

NORTH GLOUCESTERSHIRE INDEPENDENT COMMODORE PRODUCTS USERS GROUP

24 Branch Hill Rise
Charlton Kings
Cheltenham
Glos GL53 9HW
Contact: B E Syratt (Secretary)

This group was found in 1982 and, as a regional body affiliated to the national group, is able to take advantage of ICPUG facilities. Its aims are to provide a forum for discussion, exchange of knowledge and general help on anything related to Commodore computers, peripherals and software. For an annual membership fee of only £3, members receive a monthly newsletter.

Meetings are held at 8pm on the last Thursday of each month at Cheltenham Ladies' College. The staff have considerable teaching experience and an extensive collection of CBM hardware and software which allows hands-on activity at meetings.

The group is based in Cheltenham but it attracts members from all over Gloucestershire and also from neighbouring counties. For further information, contact the Secretary at the above address.

HORSHAM MICROCOMPUTER CLUB

1 Kingsmead Road
Broadbridge Heath
Horsham
West Sussex RH12 3LL
Contact: Philip Dickinson (Club Secretary)
Tel: 90403 60965

The Horsham Micro's meet every second Wednesday of the month from 7.30 to about 10.00 pm at the Forest Community School, Horsham. Machines catered for include the VIC 20, CBM 64, Dragon, Spectrum, BBC, Atari and Genie. The subscription is £10 and includes access to the School Computer Room (about ten Pets).

All new members are welcome, with or without a micro and amongst our activities is a course in BASIC every month. An added attraction could be the Community Centre bar!

NOTTINGHAM BBC MICRO USER CLUB

8 Warkton Close
Chilwell
Nottingham NG9 5FR
Contact: John Day
Tel: 0602 225660
Prestel: 602225660

This is a newly formed sub-group of the well established Nottingham Microcomputer Club. Meetings are on the second Monday of each month and the club provides a focal point in Nottingham for information about the BBC Computer, its hardware, software and sources of supply. They hope to bring like-minded users together — educationalists, radio hams, gamesplayers, businessmen and so on as well as hardware and software specialists. Why not join the club and enjoy your Beeb even more. Beginners welcome.

BELGIAN ZX MICRO CLUB

6 Boulevard Leopold III — Bte 15
B-1030 Brussels
Contact: Michel G Hunin (Secretary)
Tel: (02) 216.01.25

This club is open to all Belgian Spectrum owners. Members enjoy weekly meetings, beginners' courses in BASIC and a monthly bulletin. Any Belgians interested should contact Michel at the above address.

EASTBOURNE AND DISTRICT COMPUTER CLUB

22 Selwyn Road
Eastbourne
Sussex BN21 1LR
Contact: Bob Cooke
Tel: 0323 51437

This club meets on the fourth Wednesday in each month at 7.30pm in the WRVS Centre, Hyde Road, Eastbourne. All are welcome. Informal talks are given at meetings by a guest speaker, recently from Brighton Polytechnic and Sussex University. There has also been a demonstration of hardware and software by a local electronics firm. Other benefits include a monthly newsletter containing programming hints and tips and a BASIC teach-in.

Membership is £6.00 per annum (£3.00 for juniors) or 60p per meeting.

SHARP USERS CLUB

SCAT College
Wellington Road
Taunton
Somerset
Contact: Brian Thomas

This club is dedicated to users of the Sharp MZ-80K, MZ-80A, MZ-80B and the MZ-700 computers. It has an internal membership and a number of sub-groups who meet regularly all over the UK.

Members receive an eighty page quarterly magazine containing listings, reviews and machine code articles. The club has also fostered original hardware projects such as modems, printers and disc drive interfaces that can be built by members. They are also conducting research into voice synthesis and colour hi-res graphics.

Membership for 1984 is £7: for more information send an SAE or International Reply Coupon to the above address.

GRIMSBY COMPUTER CLUB

Contact: Ian Fell
Tel: 0472 49248 (evenings)

From its formation about three years ago, this group now has a membership of some 200 computer enthusiasts. The club offers a diverse range of activities to suit all types of people from the beginner to the confirmed 'microholic', including talks, lectures, competitions, advice, previews of new machines and trips to computerfairs around the country. The club has also organised specific user groups and runs courses such as Business Appreciation to help people realise the full potential of their micros.

Meetings are held on alternate Mondays at St James School Assembly Hall, College Street off Bargate at 7.30pm. There is an annual subscription of £8.00 (£4.00 for under 14s and OAPs) or a trial membership of £1.00 which entitles the holder to three visits and is refundable against full membership. Beginners are especially welcome as the club recognises that today's novices

are the future membership. For further details, ring Ian Fell (Publicity officer).

MERSEYSIDE NASCOM USERS GROUP

The Bungalow
Park View
Thornton
Liverpool L23 4TD
Contact: T Searle (Secretary)
Tel: 051-931 1005

This is a well established group of some eight years standing with a membership of nearly 200: predominantly Nascom owners as the name implies, but most members own a second machine such as a BBC, Acorn, Spectrum or a Gemini.

The group meets at the Liverpool Hotel, James Street on the first Wednesday of each month. Meetings start at 7.30pm and anyone calling will be made welcome. For further information, contact the Secretary at the above address.

BOLTON COMPUTER GROUP

24 Ullswater Close
Little Lever
Bolton BL3 1UD
Contact: Susy Hatton (Secretary)
Tel: 0204 792803

This club holds regular Thursday meetings in Room E5-13 Tower Block, Bolton College of HE, Deane Road Campus at 7.00pm, although the computer room is open from Monday to Friday, 7.00 till 9.00pm. There are about 150 members and the membership fee is £5.00 per person or £7.50 for family membership.

Facilities include an up to date and well-stocked computer room and regular lectures on different aspects of computing such as demonstrations by professional software houses and explanations of different languages. There are also short courses on BASIC and 6502/Z80 machine code. Why not contact Mrs Hatton for further details?

TAUNTON COMPUTER CLUB

Fir Tree Close
Back Lane
Westbury-sub-Mendip
Wells
Somerset BA5 1HZ
Contact: Dave Elliot
Tel: 0749 870479

This is a well established club in its third year. Meetings are every Tuesday during term time at the Somerset College of Art and Technology, Wellington Road, Taunton in rooms M3 and M5 between 6 and 9pm. There is a BBC Micro Econet system with ten terminals, RML 380Z and word processing facilities available for members' use as well as other machines which belong to present members. If you are interested in joining, call in at any of the meetings and have a look around or contact Dave at the above address.

BEDFORD COMPUTER CLUB

2 Sandon Close
Sandy
Beds SG19 1QT
Contact: W Thompson
Tel: 0767 82365

The renamed Bedford Computer Club now meets on the first and third Monday in each month at the Star Bar in Bedford at 7.30pm. Subscription is £5.00 pa (£2.00 if you are under 18 or in full time education). Visitors and guests need pay only 50p per meeting.

COMMODORE 64 MILANO CLUB

Via Sorrento 24
20153 Milano
Italy
Contact: Claudio Cerroni

This club was formed by a group of university students in Milan and they would very much like to get in contact with British Commodore users and user groups. If you are interested, contact Claudio at the above address.

MID KENT MICRO CLUB

65 Buckland Road
Maidstone
ME16 0SH
Contact: M K Gates (Secretary)

Formerly known as the Mid Kent TRS-80 Users Club, this club now accommodates all types of microcomputers. Meetings are held at the Maidstone Teachers' Centre, Sittingbourne Road, approximately once a month on a Friday evening at 7.00pm.

Any enquiries should be made to the above address.

If you own a TRS-80 and would like to exchange news, views and ideas with users in Denmark, write to Jorgen Christiansen, Stenlosevej 98, DK-2700 Bronshoj, Denmark.

Club Call,
Computing Today,
No.1, Golden Square,
London W1R 3AB.

This team would cost you about £45,000 p.a.

And they'd all like BUPA.

This team will cost you less than £1250 p.a.*

And it never goes sick.

Business accounting no longer need piles of paper, banks of filing, and a clerical army to control it all.

Just one small disc, a plain-English manual, and two hands.

Harness Pegasus Business Software to any of a dozen leading microcomputers, and watch.

At your command, Pegasus will record sales. Issue invoices. Update stocks. Process your payroll.

It can list, file . . . perform, in fact, any one of the thousand functions that today's complex business accounting demands.

All with perfect accuracy. And at a speed you won't believe until you see it.

Yet for all its pace, Pegasus puts you in control. It processes information only at the rate you want. Prompts you politely through every step in its programme. Not with complicated codes, but in simple, direct English.

With the versatile seven-module Pegasus system, you can start small, and grow. Begin, for example, with Sales Ledger.

Then build on it module by module, to match your needs.

It's not difficult to see why thousands of organisations large and small have purchased almost 20,000 modules in less than 2 years.

This is business software structured by experts sensitive to the user's needs, and programmed by professionals steeped in computer experience.

Any one of the 500 or so Pegasus Approved Dealers across the country is ready **now** to put Pegasus Software through its paces for you.

To start planning your team, simply clip and return the coupon.

The Seven-Module System
Sales Ledger, Purchase Ledger, Nominal Ledger, Invoicing/Sales Order Processing, Stock Control, Payroll, Job Costing.

*Based on lease purchase of a complete seven-module system over a basic 5 year term, ex VAT.

Pegasus Software Ltd, Freepost, Brikat House, Montagu Street, Kettering. Telephone: 0536 522822 Telex: 341297 BRIKAT G

To: Pegasus Software Ltd., Freepost, Brikat House, Montagu Street, Kettering. Telephone: 0536 522822 Telex: 341297 BRIKAT G

I would like to see Pegasus Software in action. Please send me further information and tell me where I can see the system in operation.

Name ______________ Title ______________

Company ______________

Address ______________

BLOCK CAPITALS PLEASE

Tel ______________ Telex ______________

Type of business

Our microcomputer is ☐ Sirius ☐ Victor ☐ Apricot ☐ IBM PC ☐ DEC Rainbow ☐ NCR DMV ☐ Wang PC ☐ TI Professional ☐ Canon AS-100 ☐ CBM 8000 ☐ CBM 700 ☐ HP 150 ☐ Duet-16

☐ We don't own a computer

No time to fill in the coupon? Just attach your business card.

CT/6/T

PEGASUS SOFTWARE

Harness the power!

RANDOM THOUGHTS

James I. Bartholomew

We use a modicum of statistical theory to show that, given a BASIC with a RND function, you can tailor expressions to produce any distribution of randoms you might require.

You are slowly cutting your way through the jungle when you are pounced upon by a vicious Swamp Monster. Locked in deadly combat, you draw back your sword and aim a blow at the monster's head. Your aim is true, but will you kill the monster outright or can it fight back and finally devour you? Your very life is in the balance, and it is the BASIC RANDOM function which will determine your fate.

We are all familiar with the use of RND in games which contributes that element of chance and ensures each time played is different. It is difficult to think of a game which does not use it — even advanced chess programs. There are many serious uses too, especially in simulation techniques. In fact, without the serious uses, BASIC might not have been designed with a random function as standard.

Throughout the article the convention used is that RND(0) will return a random real number between 0 and 1 while RND(n), where n is a positive integer, returns a random integer between 1 and n inclusive. All microcomputers have the first function, sometimes written as RND or RND(1), but a few do not have the second. It can be simulated by using INT(RND * n) + 1.

TEST YOUR RANDOM

Every RND(0) function will return a different number each time it is used, but is the number really random? Is the distribution of numbers what it should be? The answer to the first question is no! All microcomputers use a pseudo-random sequencer to generate numbers, and if you knew the formula it uses and the current seed value you could calculate the next number it will generate. However, if the answer to the second question is yes — the distribution is what it should be, and the next number (in practice) is unpredictable — then for almost all applications the function is good enough. A little bit of statistics will show how to check the distribution.

In drawing numbers at random from between 0 and 1, the probability of getting each number is equal. The graph of the probability distribution is given in Fig. 1. For a distribution such as this the mean value is 0.5 and the standard deviation equals $1/\sqrt{12}$ or 0.289. By calculating the mean and standard deviation of a large quantity of numbers we can test our RND(0) function.

The mean is given by XM =S/N where S is the sum of all the numbers we generate by X=RND(0) and N is the number of numbers. The standard deviation is:

$$SD = \sqrt{(\Sigma(X - XM)^2/N)}$$

where (X— XM) is the deviation of each value X from the mean. It is easier to calculate by the equivalent formula:

$$SD = \sqrt{(SS/N - XM^2)}$$

where SS is the sum of the squares of the numbers generated.

Program 1 uses these formulae to calculate the mean and standard deviation of your random number generator. The value $SD/\sqrt{N}$ is called the standard error of the mean and is an indication of how precise the mean is expected to be. The probability that the true mean lies within plus or minus one standard error of the calculated mean is approximately two thirds. In other words, if you run Program 1 several times, and approximately two out of three times the range of probable values for the mean includes 0.5, then your random function is quite good enough.

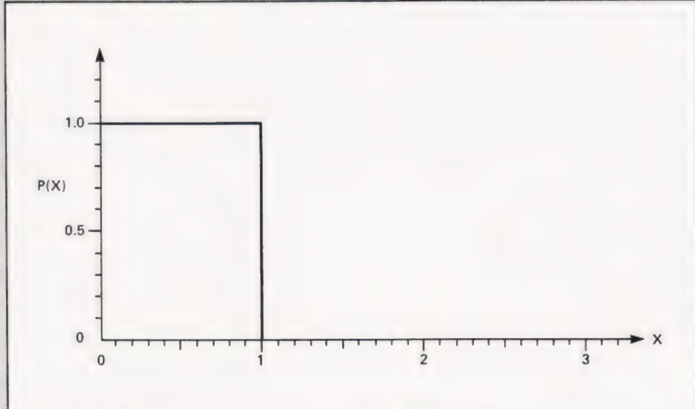


Fig. 1 Probability distribution of X = RND(0).

OTHER DISTRIBUTIONS

Most random functions give an equal probability of returning any of the numbers which it is possible to return. Sometimes, however, you might want to change the probability distribution; may be you want high numbers to be more likely than low, or perhaps you want a particular central value to be most likely with the probability reducing as you move away, either higher or lower, from the central value.

A good example of this is any board game which involves rolling two dice. Rolling a single die is equivalent to the statement X=RND(6), and rolling two dice is equivalent to X=RND(6) + RND(6). Table 1 shows all the possible combinations of two dice and Fig. 2 shows the probability distribution. A throw of 7 is the most likely combination with six ways of obtaining it, dropping down to 2 and 12 with only one way each. A double 6 should only be thrown one time out of 36, so offering an extra throw in those circumstances is quite reasonable.

Notice the triangular shape of the distribution in Fig. 2. This is more easily seen in the probability distribution for X=RND(0) + RND(0) shown in Fig. 3. This is sometimes known as the 'Triangular Distribution'. The mean value is 1 and is also the most likely value, so it is easy to convert to any other value you may wish. If you want

```
199 REM   Generate random numbers
200 S=0: SS=0: N=100
210 FOR I=1 TO 100
220      X=RND(0)
230      S=S+X                          : REM   S is sum
240      SS=SS+X*X                      : REM   SS is sum of squares
250 NEXT I
260 MN=S/N                              : REM   MN is mean
270 SD=SQR(SS/N-MN*MN)                  : REM   SD is standard deviation
280 SE=SD/SQR(N)                        : REM   SE is standard error
299 REM   Print out the results
300 PRINT "Ideal   mean = 0.5"
310 PRINT "Actual  mean =" MN
320 PRINT
330 PRINT "Ideal   std dev = 0.289"
340 PRINT "Actual  std dev =" SD
350 PRINT
360 PRINT "Probable mean from" MN-SE "to" MN+SE
```

Program 1. Test of the random function.

TABLE 1

		Second die					
	+	**1**	**2**	**3**	**4**	**5**	**6**
	1	2	3	4	5	6	7
	2	3	4	5	6	7	8
First	**3**	4	5	6	7	8	9
die	**4**	5	6	7	8	9	10
	5	6	7	8	9	10	11
	6	7	8	9	10	11	12

Possible results for two dice.

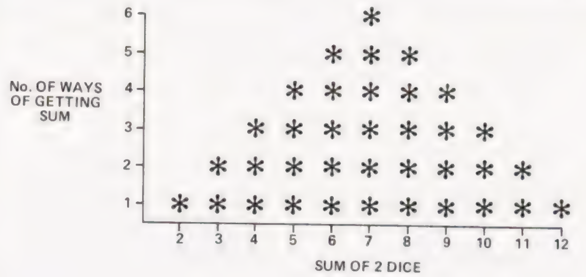


Fig. 2 Probability distribution for two dice.

a most likely value of 50 dropping off to zero at 0 and 100 then use X = 50 * (RND(0) + RND(0)).

If you use X = RND(0) − RND(0) then a triangular distribution is again obtained, but with a mean and most probable value at 0 dropping to zero at − 1 and + 1. So, for example, if you want a most probable value of 50 dropping to zero at 40 and 60 then use X = 10 * (RND(0) − RND(0)) + 50. If only integer values are required then use X =RND(10) − RND(10) + 50.

If you want the most probable value to be at one end of the range rather than the middle then try X = ABS(RND(0) − RND(0)) which gives the distribution in Fig. 4. The lower the value the more probable it is. The opposite slope can be achieved by X = 1 − ABS (RND(0) − RND(0)) and in this case 1 is the most probable value.

As an example, a function which will generate a random number from 1 to 10, and the lower the number the higher the probability of achieving it, is X =ABS (RND(10) − RND(10)) + 1.

NORMAL BEHAVIOUR

One of the most useful probability distributions, occurring frequently in statistics and engineering, is known as the Normal Distribution. It has a distinctive bell-shaped curve and its importance was stressed by A.C. Aitken — "The role of the Normal Distribution in statistics is not unlike that of the straight line in geometry." Figure 5 is a graph of a normal distribution.

Such things as the distortion of heights in the population, the weights of bags of sugar and

Program 2. Generation of normally distributed random numbers.

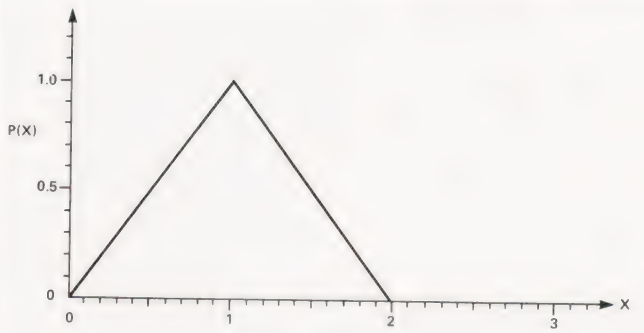


Fig. 3 Probability distribution for X = RND(0)+RND(0).

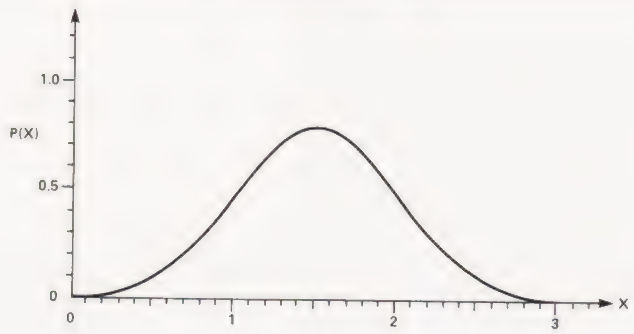


Fig. 5 Normal distribution with $\mu = 1.5$ and $\sigma = 0.5$.

Fig. 4 Probability distribution for X = ABS (RND(0) — RND(0)).

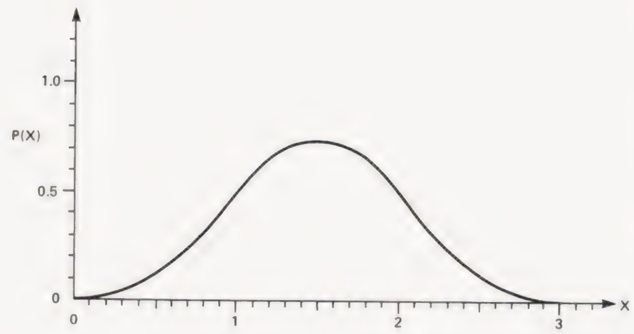


Fig. 6 Probability distribution of X = RND(0)+ RND(0)+RND(0).

TABLE 2

X = RND(0)	
$P(X) = 1$	$: 0 < X < 1$
$= 0$	: elsewhere
X = RND(0) + RND(0)	
$P(X) = X$	$: 0 < X < 1$
$= 2 - X$	$: 1 < X < 2$
$= 0$	: elsewhere
X = RND(0) + RND(0) + RND(0)	
$P(X) = 0.5X^2$	$: 0 < X < 1$
$= -X^2 + 3X - 1.5$	$: 1 < X < 2$
$= 0.5(3 - X)^2$	$: 2 < X < 3$
$= 0$	: elsewhere
Normal Distribution	
$P(X) = (1 / \sigma\sqrt{(2\pi)}) * EXP(-(X - \mu)^2 / 2\sigma^2)$	

Equations of probability distributions

levels of noise superimposed on electronic signals all tend to be normally distributed. Sometimes, especially in simulations, it is useful for a programmer to be able to create normally distributed random numbers. An approximation to this which is good enough for many applications is X = RND(0) + RND(0) and this distribution is shown in Fig. 6.

Comparison with the normal distribution in Fig. 5 is instructive, as both distributions have a mean of 1.5 and a standard deviation of 0.5. Visually the approximation looks good. The peak value for the normal distribution is 0.798 compared with 0.750 for our approximation. The approximation can only produce numbers between plus and minus 3 standard deviations of the mean, but as 99.73% of all values lie between these limits in a normal distribution this is not a serious problem.

If we multiply our new random numbers by a constant factor then the mean and standard deviation will also be multiplied by that factor. For example, if we use X = (RND(0) + RND(0)) * 10 then the mean equals 15 and the standard deviation 5. If we want a standard deviation of 20 we must multiply our numbers by 40.

By adding a constant difference to our number we can change the mean without affecting the standard deviation. So, using our previous example, X = (RND(0) + RND(0) + RND(0)) * 10 + 10 gives a mean of 25 but still a standard deviation of 5.

Program 2 is a short demonstration program which shows how you can generate your own random numbers with your desired mean and standard deviation.

FURTHER POSSIBILITIES

In general, the more RND(0)s you add together the closer will be the approximation to a normal distribution. So if three RND(0)s are not good enough for you then use more, bearing in mind the time penalty incurred if you use the function a lot of times.

If you add together n RND(0)s then the mean value will be n/2 and the standard deviation will be $\sqrt{(n/12)}$. Adding six RND(0)s will give a mean of 3 and a standard deviation of 0.707, while 12 RND(0)s give a mean of 6 and standard deviation of 1.

MATHEMATICAL NOTES

A point to bear in mind when dealing with continuous probability distributions is that P(X) is not the probability of obtaining a particular value X. Strictly speaking the probability of obtaining any particular value is zero.

The true value of probability is given by the area under the curve. The total area under the curve for the range of all possible values of X should equal 1. The probability of X being between any values a and b equals the area under the curve from X = a to X = b.

For example, if X = RND(0) then the probability that X is between 0.25 and 0.75 is 0.5. The probability of it being in the same range when X = RND(0) + RND(0) is 0.25.

The equations of probability distributions are as given in Table 2.

DATA GENIE

Peter Rabett

The ZX Spectrum is a very good machine for playing arcade games on. However, there are those who attempt to press it into worthier service. Here we look at a database package from Audiogenic.

Sir Clive Sinclair, like Oscar Wilde, will go down in history immortalised by some famous witty epigrams. "Twenty-eight days delivery" is one which still has them rolling in the aisles (unless you're a customer, of course), and another is the one about being able to control a power station with his machines. The latter point is debatable, but software houses are continually bringing out packages to put the Spectrum on a more business-like footing. There are word-processors and spreadsheets, and also databases. One such is the Data Genie from Audiogenic, described as a home database.

ON THE MENU

The novel thing about the Data Genie is its method of controlling the program operation. Everything except the entry of actual text into the database is performed using menus, and indeed by only three keys — two to move the menu cursor up and down, and Enter to choose an option. For menu selection only, key 8 may be used instead of Enter, and since 6 and 7 control the menu selection, program control is reduced to a simple three-finger exercise. It's quite like playing an arcade game!

Once the program has loaded, which takes about two and a half minutes, the user is asked to enter the maximum permissible length of field entries. This can be between 10 and 20 characters, and determines the maximum number of records that can be stored. Then the main menu appears (Photo 1). The border is coloured, and a coloured bar marks the menu option currently selected.

The main menu options allow you to set up the file structure, enter data into the records, store the database on tape, clear the screen (you'll see why soon), print records out on the screen or the ZX Printer, exit from the database, choose between alternative character sets (Sinclair's as in Photos 1-6, or the Genie's, Photos 7 and 8), or reset the system. This latter clears out the memory and allows you to start a new file with a different field structure.

Let's start by choosing SETUP FILE and pressing Enter. The screen now looks like Photo 2. The original menu is still there but it's been overwritten by a secondary menu relating to the option selected. All choices are made like this in the Genie, with other sub-menus appearing, all in different shapes, colours and positions on the screen. This explains the Clear Screen option on the main menu — after many menu operations the screen can become very untidy, so CLS wipes the slate and redisplays just the main menu.

To set up the fields, you first select the number in each record. This brings up another menu (Photo 3) which allows you to enter the number. In our example this is seven (the maximum is 15). Exiting this menu brings back the SET UP FILE menu to the foreground (Photo 3), whereupon selecting ENTER FIELD allows the names of the individual fields to be allocated, such as name, company, address, phone number and so on. If you decide you need more or fewer fields later on, it is possible to call up a secondary menu beside the ENTER FIELDS menu (Photo 5), and using this you can move the cursor in the FIELDS block to the correct position and then insert or delete as required. Fields are renamed simply by over-typing the name.

ENTERING DATA

Photo 6 shows the screen after a CLS and ENTER RECORD selection. This shows the current record being accessed, the total number of records and the maximum number possible (between 73 and 146 depending on field length). When entering a record, the screen goes blank and a flashing cursor appears at the bottom where you enter your data. A bar over the input line reminds you of the maximum length of the data input which you chose.

It's good to stick with a standard system of data entry — all lower case, all capitals, initial capitals only and so on. This is because if you enter "Genie" and then search for "GENIE", the database won't find it.

VIEWING AND EDITING

Once the records are entered, they can be looked at using the VIEW option in Photo 6. The screen blanks and displays the current record as shown in Photo 7; a file on CT's editor. A small menu also appears which can be used to edit the fields, step backward or forward through the records, or edit or list a specific record (by entering its record number).

If the menu obscures any of the record, you can delete it by selecting LOOK, which redisplays the record without the menu: this is brought back by pressing any key.

Whole records may be deleted — either the current one, a numbered one, or the whole lot — by using the menu option DELETE RECORDS.

STORAGE

Calling the STORAGE option on the main menu, which we've returned to in Photo 8, allows tape operations to take place. Records may be saved on tape, verified and loaded back in. The Name option on this menu refers to the filename to be used for tape storage, and DATA is the default. You can change it to anything with eight characters or less.

If a tape loading error occurs during verification or loading, the program breaks out with an error message. All you have to do is type in GO TO 10, which restarts the program without corrupting any of the data.

For some strange reason there is no Microdrive option on the Genie. Perhaps Audiogenic distrust them as much as I do.

SEARCHING

The search option is what makes the Data Genie more than just a complicated electronic address book. The SEARCH option lets you enter a string and displays the first record it finds that contains that string (in any field), together with the name of the field that contained the string. The second option, CONT SEARCH, is used if the first occurrence is not the one you were looking for. It will find the second, third (and so on) occurrences of the string, but you have to type in the whole string each time. This is a major fault in the program: the search option should default to the most recently specified string to prevent a great deal of unnecessary typing.

Remember also the point made earlier about the use of upper and lower case letters. If you get the case wrong, the search will prove negative.

CONCLUSIONS

The presentation of the Data Genie is nice, but a trifle gimmicky. You can only read one menu at a time, so why not just clear the screen each time and avoid the overlaying clutter? It almost seems as if the author was trying to simulate the operation of windows on a computer that uses a mouse, which is actually a different type of operation altogether.

I wonder how useful the

Photo 1

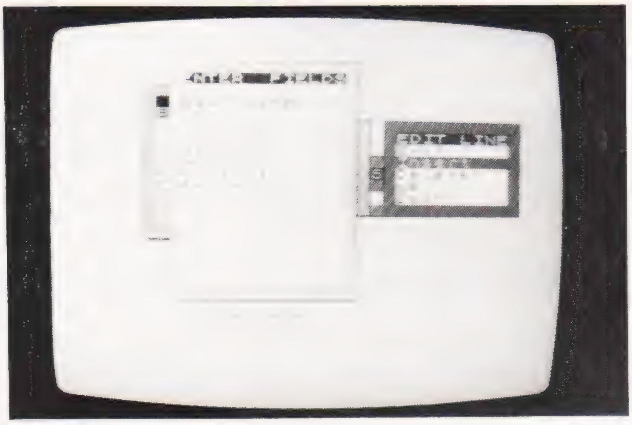


Photo 5

Photo 2

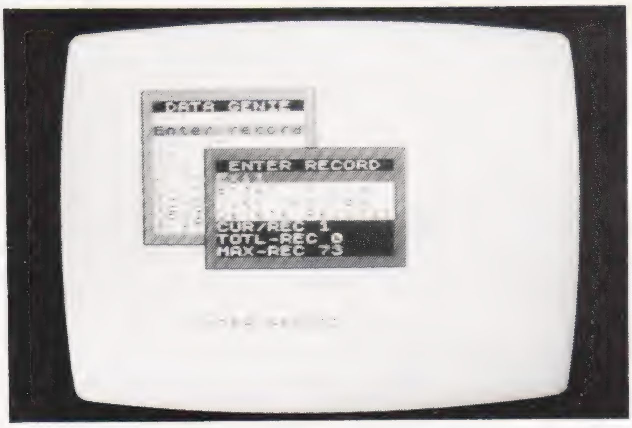


Photo 6

Photo 3

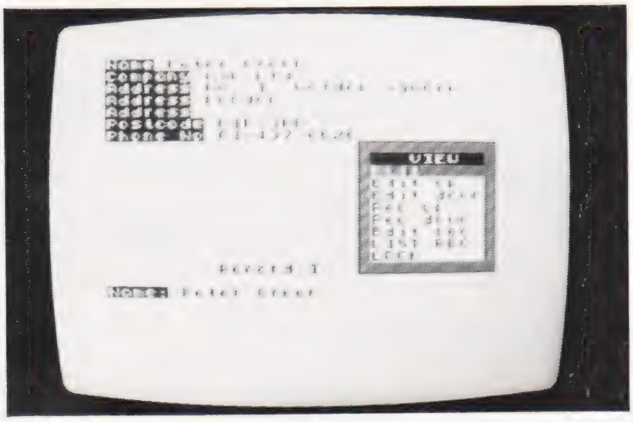


Photo 7

Photo 4

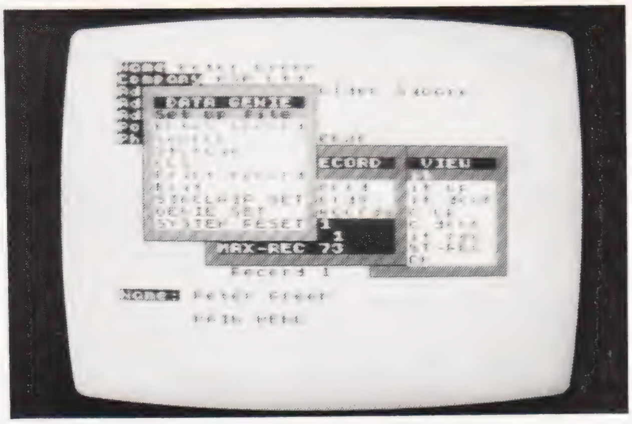
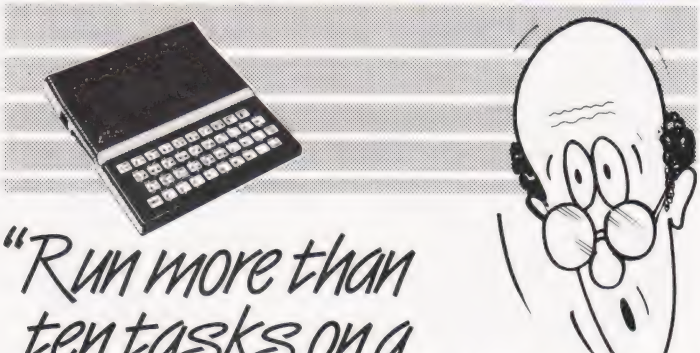


Photo 8

Spectrum is for this type of program, too. The keyboard is not adequate for typing in large amounts of text, and no mass storage other than tape is catered for. I cannot see a business, even a small one, being run with this system, and at a home a notepad is probably almost as much use.

In short, nice software, well-written, but without much potential on the Spectrum. Magpie, a similar program from Audiogenic for the Commodore 64, uses disc storage and will be considered in a future article on Commodore 64 business software.



LUCAS MICRO

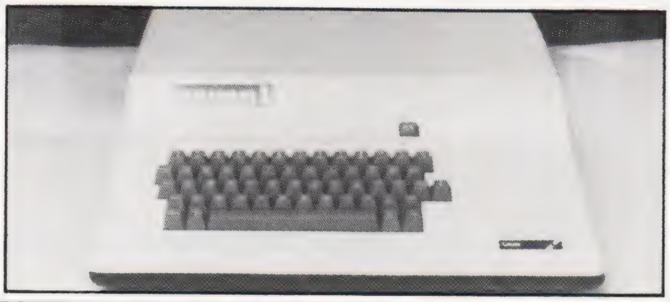
LUCAS LX

MEMORY	64K RAM expandable to 256K
LANGUAGE	Microsoft BASIC
CASSETTE	300 or 1200 baud
DISC	Single or twin 5¼ floppy disc drives DOS CP/M 2.2 (supplied) or NAS-DOS
KEYBOARD	QWERTY ☑ CURSOR ☑ NUMERIC ☑ FUNCT ☑
DISPLAY	TV ☑ MONITOR ☑ SUPPLIED ☑
INTERFACE	PARA ☑ SERIAL ☑ BUS ☑
GRAPHICS	BLOCK ☑ USER ☑ LINE ☐ RES 392 by 256 COLOUR 8 TEXT 80 by 25

Notes. The Lucas LX is a Z80A microcomputer aimed more at the professional and business user. Hence 5Mb Winchester disc interfacing is provided. Popular printers may be used with the RS232 serial interface, and a Centronics interface is also provided. There is an additional parallel interface connector for providing up to 16 on/off signals. The monitor supplied as standard is a 12" monochrome version: a colour monitor is also available. The high res colour graphics may be 392 by 256 in eight colours, or 784 by 256 in two colours. A wide range of applications software is available via the CP/M operating system, including Wordstar, Supercalc, and Calcstar.

NASCOM 3

MEMORY	48K RAM 14K ROM
LANGUAGE	Microsoft BASIC
CASSETTE	300 or 1200 baud
DISC	extra DOS CP/M or NAS-DOS
KEYBOARD	QWERTY ☑ CURSOR ☐ NUMERIC ☐ FUNCT ☐
DISPLAY	TV ☑ MONITOR ☑ SUPPLIED ☐
INTERFACE	PARA ☑ SERIAL ☑ BUS ☑
GRAPHICS	BLOCK ☑ USER ☑ LINE ☐ RES 784 by 256 (two colours) 392 by 256 (four colours) COLOUR 8 TEXT 25 by 80
SOUND	optional



DEALER CBM MICRODEALER

CHESHIRE

Automated Business Equipment Ltd — Stockport — 061-432 0708
CBM specialist for industrial accounts, management graphics and planned maintenance.

HERTFORDSHIRE

Alpha Business Systems Ltd
Church Street, Industrial Area, Ware, Herts. Tel: 0920 68926
Specialists in recommending and supplying complete systems for small businesses.

LANCASHIRE

COMMODORE BUSINESS SYSTEMS
Commodore Home Computers Software and Hardware
01 - 228 1637
2/4 Oxford Road, Manchester M1 5QA (Opposite BBC)

LONDON

MAYFAIR MICROS
a full range of Commodore uipment at very competitive ces. **5th Floor, 65 Duke Street, London W1. Tel: 01-629 2487**

COMPUTER CABIN
e store with everything for the enusiast. Official Commodore Dealer and service and information centre
24 The Parade, Silverdale, Newcastle, Staffs. Tel: 0782 636911.

TYNE & WEAR

KEY COMPUTER SERVICES LIMITED
icro Computer solutions to everyday business problems.
Osborne House, 28 Osborne Road, Newcastle upon Tyne NE2 2AJ.
Telephone: (0632) 815157
Directors: J. Sowerby, V. Shepernson.

WALES

SIGMA SYSTEMS LTD
266 North Road, CARDIFF
Tel: 621414
Also authorised dealer for CBM Digital, IBM, ACT, Sirius & Apricot

WEST MIDLANDS

MICRO BUSINESS CENTRE LTD
Wolverhampton Computer Centre, 17-19 Lichfield Street, Wolverhampton, West Midlands WV1 1EA. Tel: (0902) 29907 or 29021
Complete range of Commodore products always available.

YORKSHIRE

YORKSHIRE ELECTRONICS
Commodore Appointed Commerical Systems Dealer
Full range of peripherals and Software available
Caxton House, 17 Fountain St., Morley, West Yorkshire.
Tel: 0532 522181

DEVON

PLYMOUTH'S DEALER nascom
S & R BREWSTER LIMITED
86-88 Union Street, Plymouth PL1 3HG
Tel: 0752 665011 Open: 6 days

DORSET

PARKSTONE COMPUTER SYSTEMS
Main Nascom Dealer & Service Centre
Also Dealers for Gemini, Transtec, Genie
18 Station Road, Lower Parkstone, Poole, Dorset BH14 8UB.
Tel: PARKSTONE (0202) 746555

ZIPPY ELECTRONICS
West Dorsets Nascom Dealer
Business and Personal Computer Systems, Printers, Software. Tel: Bridport (0308) 56539
Mail Order Service operated
Access/Barclaycard accepted

LANCASHIRE

EV COMPUTING
700 Burnage Lane
Manchester M19
Tel: 061-431 4866
80-BUS SOLUTIONS

SUFFOLK

M. D. W. ELECTRONICS
47 Woodbridge Rd. East, Ipswich IP4 5QN.
Tel: (0473) 78295
Nascom, Nasbus, Gemini, Multiboard Stockists

WALES

BYTE-RITE
Llandaff R & TV Ltd, 24-26 High St., Llandaff, Cardiff. Tel: 0222 563760
Nascom, Gemini, Lucas Logic, CAD, Networks, Dragon, Peripherals

WARWICKSHIRE

BUSINESS & LEISURE MICROCOMPUTERS
16 The Square, Kenilworth, Warwickshire CV8 1EB
Tel: 0926 512127
We are the largest Lucas Nascom dealer in the UK and carry the greatest variety of products for this range of computers.

COMMODORE 720

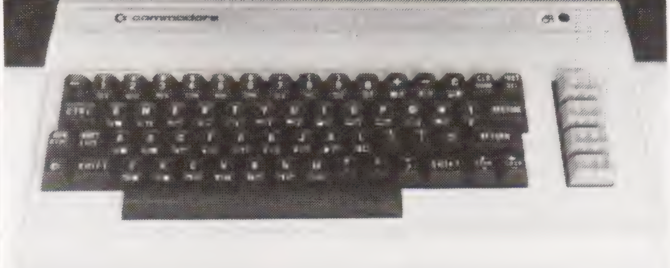
MEMORY	256K 20K ROM
LANGUAGE	Commodore BASIC
CASSETTE	300 baud
DISC	Twin in-built floppy drives
KEYBOARD	QWERTY ☑ CURSOR ☑ NUMERIC ☑ FUNCT ☑
DISPLAY	TV ☐ MONITOR SUPPLIED ☑
INTERFACE	PARA ☑ SERIAL ☑ BUS ☐
GRAPHICS	BLOCK ☑ USER ☐
	LINE ☐ RES 80 by 25
	COLOUR 16 TEXT 80 by 25
SOUND	**Three channels**

Notes. The Commodore 720 is the top model in the 700 range of business machines. It is built round the 6509 processor, but there is a dual processor (Z80 or 8088) option. The machine has been designed to meet the IEC specifications. The black-and-white monitor screen is integral and features tilt and swivel. The keyboard may be detached. The dual disc drives are built-in to the main housing and use DMA transfer, increasing speed.

COMMODORE 64

MEMORY	64K RAM 26K ROM
LANGUAGE	PET BASIC
CASSETTE	300 baud
DISC	extra DOS
KEYBOARD	QWERTY ☑ CURSOR ☑ NUMERIC ☐ FUNCT ☑
DISPLAY	TV ☑ MONITOR SUPPLIED ☐
INTERFACE	PARA ☑ SERIAL ☑ BUS ☑
GRAPHICS	BLOCK ☑ USER ☑
	LINE ☐ RES 80 by 25
	COLOUR 16 TEXT 40 by 25
SOUND	**Three channels**

Notes. The Commodore 64 is a 6510 based micro that can also use Pascal, COMAL, LOGO, FORTH and PILOT. Programs can be loaded from cassette recorder or disc drives, both extra, or cartridges. The various peripherals include printer, joysticks and games paddles.



SHARP MICRODEALER

SHARP MZ-80A

MEMORY	48K RAM	4K ROM		
LANGUAGE	Microsoft BASIC			
CASSETTE	1200 baud (built-in)			
DISC	extra	DOS		
KEYBOARD	QWERTY ☑	CURSOR ☑	NUMERIC ☑	FUNCT ☐
DISPLAY	TV ☐	MONITOR ☑	SUPPLIED ☑	
INTERFACE	PARA ☑	SERIAL ☐	BUS ☑	
GRAPHICS	BLOCK ☑	USER ☐		
	LINE ☐	RES 80 by 50		
	COLOUR	TEXT 25 by 40		
SOUND	Single channel			

Notes: The Sharp MZ-80A is a Z80 based micro. An expansion unit, printer, floppy disc unit and other peripherals are available. Other languages can also be used such as Pascal merely by replacing the tape. With the floppy disc option the machine can respond to higher level software such as Disc BASIC and FDOS (including BASIC compiler). A small range of business and educational software is available. The supplier is **Sharp Electronics (UK) Ltd.** Thorp Road, Newton Heath, Manchester M10 9BE.

SHARP MZ-80B

MEMORY	64K RAM	2K ROM		
LANGUAGE	BASIC (on tape)			
CASSETTE	1800 baud built-in			
DISC	extra	DOS		
KEYBOARD	QWERTY ☑	CURSOR ☑	NUMERIC ☑	FUNCT ☐
DISPLAY	TV ☐	MONITOR ☑	SUPPLIED ☑	
INTERFACE	PARA ☐	SERIAL ☐	BUS ☑	
GRAPHICS	BLOCK ☑	USER ☐		
	LINE ☑	RES 320 by 200		
	COLOUR	TEXT 25 by 80		
SOUND	3 channels			

Notes: The Sharp MZ-80B is a Z80A based micro. Various other languages can be loaded as the machine is "soft", no language being fitted in ROM. Expansion unit, the MZ-80P5 printer and the MZ-80FB floppy disc drive are also available. The supplier is **Sharp Electronics (UK) Ltd.** Thorp Road, Newton Heath, Manchester.

BERKSHIRE

Kuma Computers Limited

Fully support with software and hardware. Sharp 700, Sharp MZ80A, and MZ80B.
11 York Road, Maidenhead, Berks. Phone for details —
0628 71778

COMPUTER 100
181 Oxford House, Reading, Berks RG1 7UZ
TEL: 0734 591616
One of the UKs largest stockists

CAMBRIDGESHIRE

MS CONSULTANTS (CAMBRIDGE) LTD
Bridgefoot House, 159 High St., Huntingdon PE18 6TF. Tel: 0480 51721
Specialists in Sharp Hardware & Software

LANCASHIRE

Sumita **12-14 Avenham St. Preston (0772) 51686**
Sharp MZ-3541 Based Software. Bakehouse Orders System. Barstock System.
SALES — SERVICE SOFTWARE

LINCOLNSHIRE

SHARP CENTRE
16 Melville Street, Lincoln.
Tel: Lincoln 32379
All Sharp Microcomputers. Specialists in Industrial Process Control Systems. Customers include British Telecom, M.O.D. CRA & CTA Members

LONDON

SHARPSOFT LTD.
With the Sharp user in mind Sharp MZ 700 including cassette £199.99
Sharpsoft Ltd, Crisallen House, 86-90 Paul Street, London EC2.
Tel: 01 - 729 5588.

MICROS FOR MANAGERS
149 Gloucester Rd., London SW7 4TH
New program for PC1251: cost estimatic and stock additive. PC5000 stock £1639. Ask us for portable and pocket computer hardware and software.
Tel: 01-370 5125 for appointment

MIDLANDS

JAXREST LTD
Linton House, Catherine Street, Aston, Birmingham.
Tel: 021 328 0543
Sharp Service and Support

NORFOLK

GRASSROOT COMPUTERS
25 Wellington Rd., Dereham.
TEL: 0362 4640
Specialist in small business computing. Send for details of stockbox — the complete and thorough stock system with 2,500 items per disc. Other modules to follow.

SCOTLAND

micro change ltd
5 Annfield Place, Glasgow C31 2XN
Tel: 041 - 554 7623
SHARP ALTOS SUPERBRAIN
Call us for your software and hardware requirements — We will tailor our data base to meet these

LOWDATA SYSTEMS
A division of Lowdon Bros & Co (Engineers Ltd). PO Box 53, Blackness Road Dundee. DD1 9JG
Tel: (0382) 22314
Full range of Sharp Equipment stocked

WALES

STEVE'S COMPUTER COMPANY LTD
CASTLE ARCADE, CARDIFF.
Tel: 0222 — 371578
OPEN: Mon — Sat 9 - 5.30
For Components to Computers

WEST SUSSEX

MICROCENTRE LTD.
SYSTEMS, SOFTWARE SERVICE, SUPPORT
28-30 Station Road, Bognor Regis, Sussex.
TEL: 0243 827779

YORKSHIRE

PROGRAM 1
For Business Software
35 Albion Street, Hull.
TEL: 0482 20022

COMPUTAMART

AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...

BERKSHIRE

Micro General PRINTER SPECIALISTS
Call for advice on printer selection
Epson & Microline stockists. Interfaces & cables available for all popular micros
6 THE BIRCHWOODS, TILEHURST, READING.
TEL: 0734 25226

CHESHIRE

northern computers
Churchfield Rd., Frodsham.
Tel: (0928) 35110
Open: 6 days 9-6. Retail and Wholesale. Apple II & III, Atom, BBC, VIC20, 64, Newbrain, Dragon 32, Electron, Spectrum. All accessories. Easy parking off M56.

Computer Junk Shop
We Buy, Sell, Break Computers & Peripherals
10 Waterloo Rd. Widnes. Halton Tel: 051 420 4590

CORNWALL/DEVON

A. B. & C. COMPUTERS (CT)
Duchy House, 6 Lower Aylmer Sq., St. Austell.
Tel: 0726 64463/67337
Wide range of popular Micros, Printers, books and accessories. We stock all U need — try us first for service and competitive prices

HERTFORDSHIRE

NEWBRAIN & SANYO
HARDWARE & SOFTWARE
Printers, Epson, KDC, Juki etc. Monitors, Tape Recorders, Books, Expansions, CP/M. **Sayno 550/555 Computers.** Access/Mail Order. Ask for details.
ANGELA ENTERPRISES
Tel: Stevenage (0438) 812439 anytime

LANCASHIRE

LANCASHIRE MICROS
51 QUEEN STREET, MORECAMBE.
Tel: 411435. Also open Sundays.
Sinclair, Commodore, Acorn, BBC + Dragon. Authorised dealer for Bug-Byte, Imagine, Quicksilva, Artic, Melbourne House, Silversoft etc.

NSC COMPUTER SHOPS
29 Hanging Ditch, Manchester.
Tel: 061 832 2269
Open: Mon-Fri 9.30am-5.30pm.
Sat 10-5. Retail and Wholesale.

TO FILL THIS SPACE RING IAN ON 01-437 0699

LINCOLNSHIRE

SHARP CENTRE
16 Melville Street,
Lincoln.
Tel: Lincoln 32379.
Open: 9am-5.30pm closed Wed.

LONDON

HENRY'S COMPUTER SHOP
404-406 Edgware Road, London W2 1ED.
Tel: 01-402 6822
Open: 6 days a week. Order by phone or call in and see for yourself

A. J. Duchesne
(Computer Consultants) Limited
Specialists in Small Business Computing
10-12 Creechurch Lane, London EC3A 5AY
Telephone: 01-621 0433

COMPUTING +
264 Earls Court Road, London SW5. Tel: 01 373 4508 or 01 373 5000 Ext 240 Suppliers of mini and micro computers for home and office use, together with supporting software. Leading manufacturers' products supplied

LOOKING FOR MICRO-COMPUTER HARDWARE OR SOFTWARE?
THEN LOOK NO FURTHER THAN COMPUTAMART

SOUTH LONDON

CROYDON COMPUTER CENTRE
Authorised Acorn Service Centre
29a Brigstock Rd., Thornton Heath, Surrey. Tel: 01 - 689 1280
BBC, Acorn, Electron, Genie, Oric, Kaga Microvitek Zenith Monitors OKI 80, 82A - 84 Printers. Paper Ribbons. Software etc. BUY-HIRE

MIDDLESEX

L.B. ELECTRONICS
11 Hercies Rd, Hillingdon.
Tel: Uxbridge 55399 (24hr ans. service)
Open: 6 days, 9.30am-6pm, (lunch 1-2.15 except Sat) Surplus equipment, memory, EPROMs etc. Also established mail order service.

NORFOLK

ANGLIA COMPUTER CENTRE
88 St Benedicts Street, Norwich.
Tel: (0603) 29652/26002.
Open: 6 days 9am-5.30pm.

NORTHAMPTONSHIRE

NORTHAMPTON
HOME COMPUTER CENTRE
58A Wellingborough Road, Northampton.
Tel: (0604) 22539
Open: 6 days a week from 10 - 6

NORTHERN IRELAND

NEWBURN Ballycorry Co. Antrim
6 Days until 8pm. Gemini, Galaxy, Nascom, Acorn Computers, Discs, monitors, printers, servicing, hire, Industrial Control, Accounts, Word Processors etc.
WHITEHEAD 78330

SCOTLAND

VICTOR MORRIS GLASGOW
TANDY TRS 80, VIC 20, VIDEO GENIE, APPLE PANASONIC, CUMANA, EPSOM ETC.
340 Argyle Street, Glasgow G2: 041 221 8958

SUSSEX

GAMER
24 Gloucester Road, Brighton.
Tel: 0273-698424.
Open: Mon-Fri 10am-5.30pm,
Sat 9am-5.30pm.

TYNE AND WEAR

HCCS ASSOCIATES
533 Durham Rd., Low Fell,
Gateshead. Tel. Newcastle 821924.
Open: 6 days 9am-5.30pm (Sat 10am-5.30pm). Specialists in: Acorn, BBC, Video Genie, VIC 20.

WALES

STEVE'S COMPUTER COMPANY LTD
CASTLE ARCADE, CARDIFF
Tel: 0222 — 371578
OPEN: Mon — Sat 9 — 5.30
FOR COMPONENTS TO COMPUTERS

SIR Computers Ltd.
91 Whitchurch Road, Cardiff.
Tel: 0222 - 21341
BBC ACORN COMPUTER

LOOKING FOR MICRO-COMPUTER HARDWARE OR SOFTWARE?
THEN LOOK NO FURTHER THAN COMPUTAMART

COMPUTING TODAY

Lineage: 40p per word.
Semi display: £9.00 per single column centimetre
Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699

Send your requirements to:
IAN ATKINSON
ASP LTD, 1 GOLDEN SQUARE,
LONDON W1

ACCESSORIES

BLANK CASSETTES!

TOP QUALITY PROFESSIONAL BRAND COMPUTER/AUDIO CASSETTES AT BUDGET PRICES

Packed in boxes of 10 cassettes complete with labels, inlay cards and library cases. Prices include VAT post & packing

Length	Box Price (10)	Qty	Amount
5 min	£4.35		
10 min	£4.90		
12 min	£4.95		
15 min	£5.00		
30 min	£5.20		
60 min	£5.80		
90 min	£7.50		

Cheque/Postal Order enclosed for £
NAME ....
ADDRESS ....
....

PROFESSIONAL MAGNETICS LTD
Dept CT, Cassette House,
329 Hunslet Rd., Leeds.
Tel: (0532) 706066

MRL PRINTERS

Chinwa CP80 195 (224 inc VAT). Star Gemini 10X (120 cps 80 col) 215 (247 inc VAT). Star Gemini 15X (120 cps 132 col) 335 (385 inc VAT). Star Delta 10 (160 cps 80 col) 359 (413 inc VAT). Juki Daisywheel (18 cps) 369 (424 inc VAT). Plus paper, ribbons & refills etc. Free delivery by Securicor.

RING 0506 31605 FOR DETAILS

MRL Printer Interfaces

Available to fit most micros: Electron, Spectrum, Atari, Commodore 64 incl free cable, free delivery. From **£39.95** inc VAT.

RING 0506 31605 FOR DETAILS

Selection of Disc Drives at bargain prices from **£135** (inc VAT) to fit BBC, Tandy, Nascom etc.

MICRO RESEARCH (Freepost)
8 Napier Square, Houston Industrial Estate, Livingstone, West Lothian EH54 5DG

Quality Cassettes/Disks/Paper

5 x C15 Cassettes	**£2.60**
10 x C15 Cassettes	**£4.90**
5 x C20 Cassettes	**£3.00**
10 x C20 Cassettes	**£5.50**

5¼ ssdd Diskettes **£1.40** each. Listing Paper (11" x 9.5") 500 Sheets only **£5.00.** Labels, Cases & Postage Inclusive. Cheques, PO's to:
C & S COMPUTER SERVICES,
144 Sutcliffe Avenue, Grimsby, South Humberside DN33 1AP.

ADD-ONS

ANIMATION GRAPHICS BOARD FOR NASBUS/Z80 BUS SYSTEMS

11½" x 8" PCB WITH NASBUS EDGE CONNECTOR CONTAINING:

• Video Display Processor TMS 9928A/29A • 16K screen RAM driven by the VDP chip • 256 x 192 pixal resolution in 16 colours • Backdrop, pattern plane and 32 'Sprite' planes • 32 'Sprites' 8 x 8, 16 x 16 or 32 x 32 pixal objects • Sprites magnified by bit in VDP registers • X,Y co-ordinates set by two bytes/sprite • 3D 'Eclipse' effect of sprite super-imposition • coincidence flag detects sprite collisions • 8 channel 8 bit A-D on board (joysticks etc.) • Twin sound generators for stereo sound • 2K Bytes C-Mos battery backed up RAM for non volatile storage • Choice of 2 real time clocks + calendar • CTC for sound generator programming and timing functions •

Kit Comprises
All functions mapped as i/o ports PCB, i/o decode prom, construction manual, 60 page software manual, tape with test and demonstration software all for £33.80 + £1.75 post and package. Colour difference Matrix Board for linear and TTL monitors PCB £8.80 + 50p post and package.

TANELORN SYSTEMS
Bank End, Micklethwaite, Bingley, West Yorkshire BD16 3JR 0274 568380

VIC20 CBM 64 OWNERS!!!

Buy direct from the manufact-urers and save pounds!!!

3 Slot Swichable Motherboards
VIC20 **£18.95★★★**
4 Slot Switchable Motherboards
VIC20 **£21.50★★★**

NEW NEW NEW

ROBOT DRIVE INTERFACE: Plugs into Vics or 64S user port and gives complete computer control of DC motors. Contains all the circuitry for stop start and reverse. Comes complete with software cassette giving full instructions and control program **£21.50**

L. W.Staines & Co.,
Unit 2, Roding Trading Estate, London Road, Barking, Essex IG11 8BU Tel: 01-591 2900

SERVICES

ELIMINATE FAULTY CASSETTES

DataClone is the first company in the UK established specifically for the duplication of data cassettes.

All other duplicating houses are audio orientated — only DataClone has a duplicating system designed from scratch purely to handle computer information.

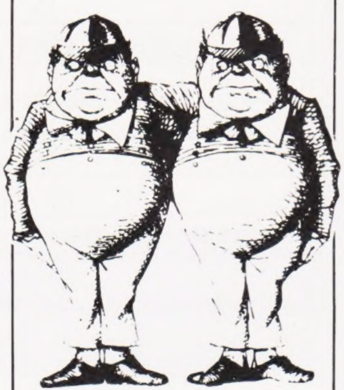
The result?

Greatly improved reliability in data transfer rates from 300 to beyond 2400 baud — previously unattainable.

All formats catered for. Quantities from 100 to infinity.

Contact us now for brochure.

DataClone — the first specialist service for computer cassettes.



DATACLONE

Unit 1, Roslin Square, Roslin Rd., Acton, London W3.
Tel: 01-993 2134 Telex: 21879

SHARP MZ-700 SERIES

THE LARGEST SELECTION OF PROGRAMS IN THE U.K. FROM THE MAJOR SHARP SOFTWARE HOUSES

From GAMES to BUSINESS and EDUCATIONAL . . . From LANGUAGES to UTILITIES
OVER 180 TITLES plus Peripherals and Accessories.

Accessories include: Matrix and Daisy Wheel Printers, Disc Drives, Joysticks, Interfaces, Cables, 80 column Mod, Paper, Colour Pens, Cassettes, Dust Covers and Books on the 700.

For your copy of THE COMPLETE MZ-700 CATALOGUE Send P.O. for 50p (refundable on first order) to:
G. M. Services (CT), D. C. Brennan Eng.,
14 North Western Avenue, Watford, Herts.

PERSONAL

FOR MEN! VASECTOMY

FREE DETAILS

Everlasting alternative to the Pill.
No fuss. No waiting lists. One visit. Low cost. In ...

BIRMINGHAM	LONDON	PLYMOUTH
GLOUCESTER	MANCHESTER	SOUTHAMPTON
LEEDS	NEWCASTLE	SWANSEA
LIVERPOOL	NORWICH	

Write today. Booklet by return. Plain cover. Or phone

LONDON	**01-388 2585**
LEEDS	**0532 440685**
MANCHESTER	**061 832 4260**

Name
Address
CT2

Marie Stopes House,
108 Whitfield St., London W1P 6BE
Caring clinics since 1925

MARIE STOPES

SEND NOW

FOR SALE

TELETYPES ASR33 £30 each, collected, or £150 for all six. Gilinsky, 15 Thornhill Park, Sunderland. 0783 44770.

SHARP MZ80A with Seikosha printer. Excellent condition still boxed, little used. Various software and manuals. Best offer over £600. Telephone Amersham 22130.

VIDEO GENIE EG3003 16K + £100 software + £30 books, £175. Write B. Myers, 2 Cypress Walk, Sale, Cheshire M33 5LW.

SHARP MZ80A immaculate manual, Basic and demonstration tapes £300 ono. Phone 01-546 evenings only.

REACH THE RIGHT AUDIENCE AT THE RIGHT PRICE RING 01-437 0699

ADVERTISERS INDEX

ANIROG

Bongo

Enjoy the hilarious antics of the comical mouse as he collects the lost diamonds. He climbs ladders, slides down chutes, use transporter and trampolines to jump across the divide. Multi screen game with three levels of difficulty.

Commodore 64 £7.95 — Vic 20 £7.95 — Spectrum £5.50

Space Pilot

Realise your dreams of being king of the open skies. Fly your aircraft into unrelenting dog fights with enemy fighters. Prove how well you can handle your craft. Five stages of tough engagements.

Commodore 64 £7.95 — Spectrum £5.50

Flight Path 737

An advanced Pilot Trainer. Written by a flight simulator instructor and pilot. Panoramic Pilot's eye view.

Commodore 64 £7.95 — Vic 20 £7.95

Also available on Disk at £9.95

TRADE ENQUIRIES: ANIROG SOFTWARE LTD. 29 WEST HILL DARTFORD KENT (0322) 92513/8

MAIL ORDER: 8 HIGH STREET HORLEY SURREY 24 HOUR CREDIT CARD SALES HORLEY (02934) 6083

PAYMENT BY CHEQUE P.O. ACCESS/VISA 50p POSTAGE & PACKAGING £2.00 OVERSEAS